

Red Hat Enterprise Linux 6
Konfiguration des Red Hat High Availability Add-Ons mit Pacemaker

Referenzhandbuch für das High Availability Add-On für Red Hat Enterprise Linux 6

Ausgabe 1

Red Hat Enterprise Linux 6 Konfiguration des Red Hat High Availability Add-Ons mit Pacemaker

---

Referenzhandbuch für das High Availability Add-On für Red Hat Enterprise Linux 6
Ausgabe 1

Rechtlicher Hinweis

Zusammenfassung

Das Handbuch Konfiguration des Red Hat High Availability Add-Ons mit Pacemaker liefert Informationen über die Konfiguration des Red Hat High Availability Add-Ons mithilfe von Pacemaker.

Inhaltsverzeichnis

Einführung **4**
1. Dokumentkonventionen 4
1.1. Typografische Konventionen 4
1.2. Konventionen für Seitenansprachen 6
1.3. Anmerkungen und Warnungen 7
2. Feedback 7

Kapitel 1. Überblick über Konfiguration und Verwaltung des Red Hat High Availability Add-Ons . 9
1.1. Installieren der Pacemaker-Konfigurationstools 9
1.2. Konfigurieren der iptables-Firewall zum Erlauben von Cluster-Komponenten 9
1.3. Cluster- und Pacemaker-Konfigurationsdateien 9

Kapitel 2. Die pcs-Befehlszeilenschnittstelle **11**
2.1. pcs-Befehle 11
2.2. pcs-Hilfebildschirm 11
2.3. Anzeigen der unformatierten Cluster-Konfiguration 12
2.4. Speichern einer Konfigurationsänderung in eine Datei 12
2.5. Anzeigen des Status 12
2.6. Anzeigen der vollständigen Cluster-Konfiguration 13
2.7. Anzeigen der aktuellen pcs-Version 13

Kapitel 3. Cluster-Erstellung und -Verwaltung **14**
3.1. Erstellen eines Clusters 14
3.1.1. Authentifizieren der Cluster-Knoten 14
3.1.2. Konfigurieren und Starten der Cluster-Knoten 14
3.2. Verwalten von Cluster-Knoten 15
3.2.1. Stoppen von Cluster-Diensten 15
3.2.2. Aktivieren und Deaktivieren der Cluster-Dienste 15
3.2.3. Hinzufügen und Entfernen von Cluster-Knoten 15
3.2.4. Standby-Modus 16
3.3. Festlegen von Benutzerberechtigungen 16
3.4. Entfernen der Cluster-Konfiguration 18
3.5. Anzeigen des Cluster-Status 18

Kapitel 4. Fencing: Konfigurieren von STONITH **19**
4.1. Verfügbare STONITH (Fencing)-Agenten 19
4.2. Allgemeine Eigenschaften von Fencing-Geräten 19
4.3. Anzeigen gerätespezifischer Fencing-Optionen 20
4.4. Erstellen eines Fencing-Geräts 21
4.5. Konfigurieren von Storage-basierten Fencing-Geräten mit Aufheben der Abgrenzung 21
4.6. Anzeigen von Fencing-Geräten 22
4.7. Bearbeiten und Löschen von Fencing-Geräten 22
4.8. Verwalten von Knoten mit Fencing-Geräten 22
4.9. Weitere Fencing-Konfigurationsoptionen 23
4.10. Konfigurieren von Fencing-Levels 26

Kapitel 5. Konfigurieren von Cluster-Ressourcen **28**
5.1. Ressourcenerstellung 28
5.2. Ressourceneigenschaften 28
5.3. Ressourcenspezifische Parameter 29
5.4. Ressourcen-Metaoptionen 29
5.5. Ressourcenoperationen 33
5.6. Anzeigen konfigurierter Ressourcen 36

5.7. Ändern von Ressourcenparametern 36
5.8. Mehrere Überwachungsoperationen 37
5.9. Aktivieren und Deaktivieren von Cluster-Ressourcen 37
5.10. Bereinigen von Cluster-Ressourcen 37

Kapitel 6. Ressourcenbeschränkungen **39**
6.1. Standortbeschränkungen 39
6.1.1. Konfigurieren eines Opt-In-Clusters 40
6.1.2. Konfigurieren eines Opt-Out-Clusters 40
6.2. Reihenfolgebeschränkungen 41
6.2.1. Zwingende Reihenfolge 42
6.2.2. Optionale Reihenfolge 42
6.2.3. Geordnete Ressourcengruppen 42
6.2.4. Entfernen von Ressourcen aus Reihenfolgebeschränkungen 43
6.3. Relative Standortbeschränkung für Ressourcen 43
6.3.1. Zwingende Platzierung 44
6.3.2. Optionale Platzierung 44
6.3.3. Relative Standortbeschränkung für Ressourcengruppen 44
6.3.4. Entfernen von relativen Standortbeschränkungen 45
6.4. Anzeigen von Beschränkungen 45
6.5. Ressourcenruppen 46
6.5.1. Gruppenoptionen 47
6.5.2. Gruppentreue 47

Kapitel 7. Verwalten von Cluster-Ressourcen **48**
7.1. Manuelles Verlegen von Ressourcen im Cluster 48
7.2. Verlegen von Ressourcen wegen Ausfall 48
7.3. Verlegen von Ressourcen wegen Änderungen der Verbindungsfähigkeit 49
7.4. Aktivieren, Deaktivieren und Ausschließen von Cluster-Ressourcen 50
7.5. Deaktivieren von Überwachungsoperationen 51
7.6. Verwaltete Ressourcen 51

Kapitel 8. Erweiterte Ressourcentypen **53**
8.1. Ressourcen-Klone 53
8.1.1. Erstellen und Entfernen einer geklonten Ressource 53
8.1.2. Klonbeschränkungen 55
8.1.3. Standorttreue von Klons 55
8.2. Multi-Status-Ressourcen: Ressourcen mit mehreren Modi 55
8.2.1. Überwachen von Multi-Status-Ressourcen 56
8.2.2. Multi-Status-Beschränkungen 56
8.2.3. Multi-Status-Standorttreue 57
8.3. Ereignisbenachrichtigungen mit Überwachungsressourcen 57
8.4. Der Dienst pacemaker_remote 58
8.4.1. Ressourcenoptionen für Container-Remote-Knoten 59
8.4.2. Host- und Gastauthentifizierung 60
8.4.3. Ändern der standardmäßigen pacemaker_remote-Optionen 60
8.4.4. Konfigurationsübersicht: KVM-Remote-Knoten 60

Kapitel 9. Pacemaker-Regeln **62**
9.1. Knotenattribut-Ausdrücke 62
9.2. Zeit-/Datumsbasierte Ausdrücke 63
9.3. Datumsspezifikationen 63
9.4. Dauer 64
9.5. Konfigurieren von Regeln mit pcs 64

9.6. Beispiel für zeitbasierte Ausdrücke 64
9.7. Verwenden von Regeln zum Bestimmen von Ressourcenstandorten 65

Kapitel 10. Pacemaker-Cluster-Eigenschaften 66
10.1. Überblick über Cluster-Eigenschaften und -Optionen 66
10.2. Festlegen und Entfernen von Cluster-Eigenschaften 68
10.3. Abfragen der Einstellungen für Cluster-Eigenschaften 69

Cluster-Erstellung in Red Hat Enterprise Linux Release 6.5 und Red Hat Enterprise Linux Release 6.6 70
A.1. Cluster-Erstellung mit rgmanager und mit Pacemaker 70
A.2. Cluster-Erstellung mit Pacemaker in Red Hat Enterprise Linux Release 6.5 und Red Hat Enterprise Linux Release 6.6 74

Konfigurationsbeispiel unter Verwendung der pcs-Befehle 76
B.1. Erste Systemeinrichtung 76
B.1.1. Installieren der Cluster-Software 76
B.1.2. Erstellen und Starten des Clusters 76
B.2. Fencing-Konfiguration 78
B.3. Konfigurieren eines Apache-Webservers in einem Red Hat High Availability Cluster mithilfe des pcs-Befehls 79
B.3.1. Konfigurieren eines LVM-Datenträgers mit einem ext4-Dateisystem 80
B.3.2. Webserver-Konfiguration 81
B.3.3. Exklusive Archivierung einer Datenträgergruppe in einem Cluster 82
B.3.4. Erstellen der Ressourcen und Ressourcengruppen mit dem pcs-Befehl 83
B.3.5. Testen der Ressourcenkonfiguration 85

Versionsgeschichte 87

Einführung

Dieses Handbuch liefert Informationen zur Installation, Konfiguration und Verwaltung der Komponenten des Red Hat High Availability Add-Ons. Die Komponenten des Red Hat High Availability Add-Ons erlauben Ihnen das Verbinden einer Gruppe von Computern (genannt *Knoten* oder *Mitglieder*), um als Cluster zusammenzuarbeiten. In diesem Dokument bezieht sich das Wort *Cluster* auf eine Gruppe von Computern, auf denen das Red Hat High Availability Add-On läuft.

Die Zielgruppe dieses Handbuchs sollte bereits über umfassende Kenntnisse von Red Hat Enterprise Linux verfügen und die Grundlagen von Clustern, Storage und Server-Rechnern verstehen.

Weitere Informationen über Red Hat Enterprise Linux 6 finden Sie in den folgenden Quellen:

- *Red Hat Enterprise Linux Installationshandbuch* – Liefert Informationen bezüglich der Installation von Red Hat Enterprise Linux 6.
- *Red Hat Enterprise Linux Bereitstellungshandbuch* – Liefert Informationen bezüglich der Implementierung, der Konfiguration und der Administration von Red Hat Enterprise Linux 6.

Weitere Informationen über das High Availability Add-On und zugehörige Produkte für Red Hat Enterprise Linux 6 finden Sie in den folgenden Quellen:

- *Überblick über das High Availability Add-On* – Liefert einen allgemeinen Überblick über das High Availability Add-On.
- *Cluster-Administration* – Liefert Informationen zur Installation, Konfiguration und Verwaltung des High Availability Add-Ons.
- *Administration des Logical Volume Manager* – Liefert eine Beschreibung des Logical Volume Managers (LVM), inklusive Informationen zum Einsatz von LVM in einer Cluster-Umgebung.
- *Global File System 2: Konfiguration und Administration* – Liefert Informationen zur Installation, Konfiguration und Pflege von Red Hat GFS (Red Hat Global File System 2), das Bestandteil des Resilient Storage Add-Ons ist.
- *DM Multipath* – Liefert Informationen über die Verwendung des Device-Mapper Multi-Pathing-Features von Red Hat Enterprise Linux 6.
- *Verwaltung der Lastverteilung* – Liefert Informationen zur Konfiguration von Hochleistungssystemen und -diensten mit dem Red Hat Load Balancer Add-On, einer Gruppe integrierter Softwarekomponenten, die Linux Virtual Server (LVS) bereitstellen, um IP-Lasten über eine Gruppe realer Server zu verteilen.
- *Versionshinweise* – Liefert Informationen über die jeweils aktuelle Version der Red Hat Produkte.

Die Dokumentation der Red Hat Cluster Suite und andere Red Hat Dokumente stehen als HTML-, PDF- und RPM-Versionen auf der Red Hat Enterprise Linux Dokumentations-CD und online unter https://access.redhat.com/site/documentation/ zur Verfügung.

1. Dokumentkonventionen

Dieses Handbuch verwendet mehrere Konventionen, um bestimmte Wörter und Sätze hervorzuheben und Aufmerksamkeit auf bestimmte Informationen zu lenken.

1.1. Typografische Konventionen

Es werden vier typografische Konventionen verwendet, um die Aufmerksamkeit auf bestimmte Wörter und Sätze zu lenken. Diese Konventionen und die Umstände, unter denen sie auftreten, sind folgende:

Nichtproportional Fett

Dies wird verwendet, um Systemeingaben hervorzuheben, einschließlich Shell-Befehle, Dateinamen und -pfade. Es wird ebenfalls zum Hervorheben von Tasten und Tastenkombinationen verwendet. Zum Beispiel:

> Um den Inhalt der Datei **my_next_bestselling_novel** in Ihrem aktuellen Arbeitsverzeichnis zu sehen, geben Sie den Befehl **cat my_next_bestselling_novel** in den Shell-Prompt ein und drücken Sie **Enter**, um den Befehl auszuführen.

Das oben aufgeführte Beispiel beinhaltet einen Dateinamen, einen Shell-Befehl und eine Taste. Alle werden nichtproportional fett dargestellt und alle können, dank des Kontextes, leicht unterschieden werden.

Tastenkombinationen unterscheiden sich von einzelnen Tasten durch das Pluszeichen, das die einzelnen Teile einer Tastenkombination miteinander verbindet. Zum Beispiel:

> Drücken Sie **Enter**, um den Befehl auszuführen.
>
> Drücken Sie **Strg**+**Alt**+**F2**, um zu einem virtuellen Terminal zu wechseln.

Das erste Beispiel hebt die zu drückende Taste hervor. Das zweite Beispiel hebt eine Tastenkombination hervor: eine Gruppe von drei Tasten, die gleichzeitig gedrückt werden müssen.

Falls Quellcode diskutiert wird, werden Klassennamen, Methoden, Funktionen, Variablennamen und Rückgabewerte, die innerhalb eines Abschnitts erwähnt werden, wie oben gezeigt **nichtproportional fett** dargestellt. Zum Beispiel:

> Zu dateiverwandten Klassen zählen **filesystem** für Dateisysteme, **file** für Dateien und **dir** für Verzeichnisse. Jede Klasse hat ihren eigenen Satz an Berechtigungen.

Proportional Fett

Dies kennzeichnet Wörter oder Sätze, die auf einem System vorkommen, einschließlich Applikationsnamen, Text in Dialogfeldern, beschriftete Schaltflächen, Bezeichnungen für Auswahlkästchen und Radio-Buttons, Überschriften von Menüs und Untermenüs. Zum Beispiel:

> Wählen Sie **System** → **Einstellungen** → **Maus** in der Hauptmenüleiste aus, um die **Mauseinstellungen** zu öffnen. Wählen Sie im Reiter **Tasten** auf das Auswahlkästchen **Mit links bediente Maus** und anschließend auf **Schließen**, um die primäre Maustaste von der linken auf die rechte Seite zu ändern (d.h., um die Maus auf Linkshänder anzupassen).
>
> Um ein Sonderzeichen in eine **gedit**-Datei einzufügen, wählen Sie **Anwendungen** → **Zubehör** → **Zeichentabelle** aus der Hauptmenüleiste. Wählen Sie als Nächstes **Suchen** → **Suchen** aus der Menüleiste der **Zeichentabelle**, geben Sie im Feld **Suchbegriff** den Namen des Zeichens ein und klicken Sie auf **Weitersuchen**. Das gesuchte Zeichen wird daraufhin in der **Zeichentabelle** hervorgehoben. Doppelklicken Sie auf dieses hervorgehobene Zeichen, um es in das Feld **Zu kopierender Text** zu übernehmen und klicken Sie anschließend auf die Schaltfläche **Kopieren**. Gehen Sie nun zurück in Ihr Dokument und wählen Sie **Bearbeiten** → **Einfügen** aus der **gedit**-Menüleiste.

Der oben aufgeführte Text enthält Applikationsnamen, systemweite Menünamen und -elemente, applikationsspezifische Menünamen sowie Schaltflächen und Text innerhalb einer grafischen

Oberfläche. Alle werden proportional fett dargestellt und sind anhand des Kontextes unterscheidbar.

`Nichtproportional Fett Kursiv` oder ***Proportional Fett Kursiv***

Sowohl bei nichtproportional fett als auch bei proportional fett weist ein zusätzlicher Kursivdruck auf einen ersetzbaren oder variablen Text hin. Kursivdruck kennzeichnet Text, der nicht wörtlich eingeben wird, oder angezeigten Text, der sich abhängig von den gegebenen Umständen unterscheiden kann. Zum Beispiel:

> Um sich mit einer Remote-Maschine via SSH zu verbinden, geben Sie an einem Shell-Prompt **`ssh`** ***`username@domain.name`*** ein. Falls die Remote-Maschine **`example.com`** ist und Ihr Benutzername auf dieser Maschine John lautet, geben Sie also **`ssh john@example.com`** ein.
>
> Der Befehl **`mount -o remount`** ***`file-system`*** hängt das angegebene Dateisystem wieder ein. Um beispielsweise das **`/home`**-Dateisystem wieder einzuhängen, verwenden Sie den Befehl **`mount -o remount /home`**.
>
> Um die Version des derzeit installierten Pakets zu sehen, verwenden Sie den Befehl **`rpm -q`** ***`package`***. Die Ausgabe sieht wie folgt aus: ***`package-version-release`***.

Beachten Sie die kursiv dargestellten Begriffe oben — username, domain.name, file-system, package, version und release. Jedes Wort ist ein Platzhalter entweder für Text, den Sie für einen Befehl eingeben, oder für Text, der vom System angezeigt wird.

Neben der Standardbenutzung für die Darstellung des Titels eines Werks zeigt der Kursivdruck auch die erstmalige Verwendung eines neuen und wichtigen Begriffs an. Zum Beispiel:

> Publican ist ein *DocBook* Publishing-System.

1.2. Konventionen für Seitenansprachen

Ausgaben des Terminals und Auszüge aus dem Quellcode werden visuell vom umliegenden Text hervorgehoben durch sogenannte Seitenansprachen (auch Pull-Quotes genannt).

Eine an das Terminal gesendete Ausgabe wird in den Schrifttyp **`nichtproportional Roman`** gesetzt und wie folgt dargestellt:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Auszüge aus dem Quellcode werden ebenfalls in den Schrifttyp **`nichtproportional Roman`** gesetzt, doch wird zusätztlich noch die Syntax hervorgehoben:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                    struct kvm_assigned_pci_dev *assigned_dev)
{
        int r = 0;
        struct kvm_assigned_dev_kernel *match;

        mutex_lock(&kvm->lock);

        match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                      assigned_dev->assigned_dev_id);
        if (!match) {
                printk(KERN_INFO "%s: device hasn't been assigned
before, "
```

```
                        "so cannot be deassigned\n", __func__);
                r = -EINVAL;
                goto out;
        }

        kvm_deassign_device(kvm, match);

        kvm_free_assigned_device(kvm, match);

out:
        mutex_unlock(&kvm->lock);
        return r;
}
```

1.3. Anmerkungen und Warnungen

Zu guter Letzt verwenden wir drei visuelle Stile, um die Aufmerksamkeit auf Informationen zu lenken, die andernfalls vielleicht übersehen werden könnten.



Anmerkung

Eine Anmerkung ist ein Tipp, ein abgekürztes Verfahren oder ein alternativer Ansatz für die vorliegende Aufgabe. Das Ignorieren von Anmerkungen sollte keine negativen Auswirkungen haben, aber Sie verpassen so vielleicht einen Trick, der Ihnen das Leben vereinfachen könnte.



Wichtig

Die Wichtig-Schaukästen lenken die Aufmerksamkeit auf Dinge, die sonst leicht übersehen werden können: Konfigurationsänderungen, die nur für die aktuelle Sitzung gelten oder Dienste, für die ein Neustart nötig ist, bevor eine Aktualisierung wirksam wird. Das Ignorieren von Wichtig-Schaukästen würde keinen Datenverlust verursachen, kann aber unter Umständen zu Ärgernissen und Frustration führen.



Warnung

Eine Warnung sollte nicht ignoriert werden. Das Ignorieren von Warnungen führt mit hoher Wahrscheinlichkeit zu Datenverlust.

2. Feedback

Falls Sie einen Fehler in diesem Handbuch finden oder eine Idee haben, wie dieses verbessert werden könnte, freuen wir uns über Ihr Feedback! Bitte reichen Sie einen Bericht in Bugzilla ein: http://bugzilla.redhat.com/bugzilla/. Reichen Sie den Fehlerbericht für das Produkt **Red Hat Enterprise Linux 6** und die Komponente **doc-Cluster_General** ein.

Vergewissern Sie sich beim Einreichen eines Fehlerberichts, dass Sie die Kennung des Handbuchs mit angeben:

```
Configuring_High_Availability_With_Pacemaker(EN)-6 (2014-8-7T16:26)
```

Indem Sie die Kennung des Handbuchs angeben, wissen wir genau, welche Version des Handbuchs Sie vorliegen haben.

Falls Sie uns einen Vorschlag zur Verbesserung der Dokumentation senden möchten, sollten Sie hierzu möglichst genaue Angaben machen. Wenn Sie einen Fehler gefunden haben, geben Sie bitte die Nummer des Abschnitts und einen Ausschnitt des Textes an, damit wir diesen leicht finden können.

Kapitel 1. Überblick über Konfiguration und Verwaltung des Red Hat High Availability Add-Ons

Dieses Dokument liefert Beschreibungen der Optionen und Funktionen, die vom Red Hat High Availability Add-On mit Pacemaker unterstützt werden.

Dieses Handbuch dokumentiert die Verwendung der **pcs**-Konfigurationsoberfläche für die Red Hat Enterprise Linux Release 6.6 und höher.



Anmerkung

Informationen über bewährte Verfahren zur Bereitstellung und Aktualisierung von Red Hat Enterprise Linux Clustern unter Verwendung des High Availability Add-Ons und Red Hat Global File System 2 (GFS2) finden Sie im Artikel „Red Hat Enterprise Linux Cluster, High Availability, and GFS Deployment Best Practices“ im Red Hat Kundenportal unter https://access.redhat.com/kb/docs/DOC-40821.

1.1. Installieren der Pacemaker-Konfigurationstools

Sie können den folgenden **yum install** Befehl verwenden, um die Softwarepakete für das Red Hat High Availability Add-On sowie alle verfügbaren Fencing-Agenten aus dem High-Availability-Channel zu installieren.

```
# yum install pcs fence-agents
```

Die Pakete **lvm2-cluster** und **gfs2-utils** sind Teil des ResilientStorage-Channels. Sie können sie bei Bedarf mit dem folgenden Befehl installieren.

```
# yum install lvm2-cluster gfs2-utils
```



Warnung

Nachdem Sie die Red Hat High Availability Add-On Pakete installiert haben, sollten Sie sicherstellen, dass Ihre Einstellungen zur Softwareaktualisierung keine automatische Installation erlaubt. Eine Installation auf einem laufenden Cluster kann unerwartetes Verhalten nach sich ziehen.

1.2. Konfigurieren der iptables-Firewall zum Erlauben von Cluster-Komponenten

Das Red Hat High Availability Add-On erfordert, dass die folgenden Ports geöffnet sind:

- Für TCP: Ports 2224, 3121, 21064
- Für UDP: Ports, 5405

1.3. Cluster- und Pacemaker-Konfigurationsdateien

Die Konfigurationsdateien für das Red Hat High Availability Add-On sind **`cluster.conf`** und **`cib.xml`**. Bearbeiten Sie diese Dateien nicht direkt, sondern verwenden Sie stattdessen die **`pcs`**- oder **`pcsd`**-Oberfläche.

Die **`cluster.conf`**-Datei liefert die Cluster-Parameter, die von **`corosync`** verwendet werden – dem Cluster-Verwalter, auf dem Pacemaker aufgebaut ist.

Die **`cib.xml`**-Datei ist eine XML-Datei, die sowohl die Cluster-Konfiguration als auch den derzeitigen Status aller Ressourcen im Cluster abbildet. Diese Datei wird von Pacemakers Cluster Information Base (CIB) verwendet. Die Inhalte der CIB werden automatisch im gesamten Cluster synchron gehalten.

Kapitel 2. Die pcs-Befehlszeilenschnittstelle

Die **pcs**-Befehlszeilenschnittstelle ermöglicht die Steuerung und Konfiguration von **corosync** und **pacemaker**.

Das allgemeine Format des **pcs**-Befehls lautet wie folgt.

```
pcs [-f file] [-h] [commands]...
```

2.1. pcs-Befehle

Die **pcs**-Befehle lauten wie folgt.

- **cluster**

  Konfiguriert Cluster-Optionen und Knoten. Informationen über den Befehl **pcs cluster** finden Sie in Kapitel 3, *Cluster-Erstellung und -Verwaltung*.

- **resource**

  Erstellt und verwaltet Cluster-Ressourcen. Informationen über den Befehl **pcs cluster** finden Sie in Kapitel 5, *Konfigurieren von Cluster-Ressourcen*, Kapitel 7, *Verwalten von Cluster-Ressourcen* und Kapitel 8, *Erweiterte Ressourcentypen*.

- **stonith**

  Konfiguriert Fencing-Geräte zur Verwendung mit Pacemaker. Informationen über den Befehl **pcs stonith** finden Sie in Kapitel 4, *Fencing: Konfigurieren von STONITH*.

- **constraint**

  Verwaltet Ressourcenbeschränkungen. Informationen über den Befehl **pcs constraint** finden Sie in Kapitel 6, *Ressourcenbeschränkungen*.

- **property**

  Legt Pacemaker-Eigenschaften fest. Informationen über das Festlegen der Eigenschaften mit dem Befehl **pcs property** finden Sie in Kapitel 10, *Pacemaker-Cluster-Eigenschaften*.

- **status**

  Zeigt den aktuellen Cluster- und Ressourcenstatus an. Informationen über den Befehl **pcs status** finden Sie in Abschnitt 2.5, „Anzeigen des Status“.

- **config**

  Zeigt die vollständige Cluster-Konfiguration in einer für den Benutzer lesbaren Form an. Informationen über den Befehl **pcs config** finden Sie in Abschnitt 2.6, „Anzeigen der vollständigen Cluster-Konfiguration“.

2.2. pcs-Hilfebildschirm

Sie können die Option **-h** für **pcs** verwenden, um die Parameter eines **pcs**-Befehls samt einer Beschreibung dieser Parameter anzuzeigen. Beispielsweise zeigt der folgende Befehl die Parameter des Befehls **pcs resource**. Nur ein Ausschnitt der Ausgabe ist hier dargestellt.

```
# pcs resource -h
Usage: pcs resource [commands]...
Manage pacemaker resources
Commands:
    show [resource id] [--all]
        Show all currently configured resources or if a resource is
specified
        show the options for the configured resource.  If --all is
specified
        resource options will be displayed

    start <resource id>
        Start resource specified by resource_id
...
```

2.3. Anzeigen der unformatierten Cluster-Konfiguration

Obwohl Sie die Cluster-Konfigurationsdatei niemals direkt bearbeiten sollten, können Sie die unformatierte Cluster-Konfiguration mithilfe des Befehls **pcs cluster cib** anzeigen lassen.

Sie können diese unformatierte Cluster-Konfiguration in einer angegebenen Datei mithilfe des Befehls **pcs cluster cib *filename*** abspeichern, wie in Abschnitt 2.4, „Speichern einer Konfigurationsänderung in eine Datei“ beschrieben.

2.4. Speichern einer Konfigurationsänderung in eine Datei

Sie können die Option **-f** zum **pcs**-Befehl verwenden, um eine Konfigurationsänderung in eine Datei zu speichern ohne Auswirkungen auf die aktive CIB.

Falls Sie zuvor bereits einen Cluster konfiguriert hatten und es bereits eine aktive CIB gibt, können Sie mithilfe des folgenden Befehls das unformatierte XML in eine Datei speichern.

```
pcs cluster cib filename
```

Beispielsweise speichert der folgende Befehl das unformatierte XML der CIB in eine Datei namens **testfile**.

```
pcs cluster cib testfile
```

Der folgende Befehl erstellt eine Ressource in der Datei **testfile1**, fügt diese Ressource jedoch nicht der derzeit laufenden Cluster-Konfiguration hinzu.

```
# pcs -f testfile1 resource create VirtualIP ocf:heartbeat:IPaddr2
ip=192.168.0.120 cidr_netmask=24 op monitor interval=30s
```

Sie können den aktuellen Inhalt der Datei **testfile** mit dem folgenden Befehl an die CIB übertragen.

```
pcs cluster cib-push filename
```

2.5. Anzeigen des Status

Sie können den Status des Clusters und der Cluster-Ressourcen mithilfe des folgenden Befehls anzeigen.

```
pcs status commands
```

Falls Sie keinen *commands*-Parameter angeben, zeigt dieser Befehl alle Informationen über den Cluster und die Ressourcen. Sie können den Status bestimmter Cluster-Komponenten anzeigen, indem Sie **resources**, **groups**, **cluster**, **nodes** oder **pcsd** angeben.

2.6. Anzeigen der vollständigen Cluster-Konfiguration

Verwenden Sie den folgenden Befehl, um die aktuelle, vollständige Cluster-Konfiguration anzuzeigen.

```
pcs config
```

2.7. Anzeigen der aktuellen pcs-Version

Der folgende Befehl zeigt die Version von **pcs** an, die derzeit ausgeführt wird.

```
pcs --version
```

Kapitel 3. Cluster-Erstellung und -Verwaltung

Dieses Kapitel beschreibt die grundlegende Cluster-Verwaltung mit Pacemaker, einschließlich der Erstellung des Clusters, Verwaltung der Cluster-Komponenten und Anzeigen des Cluster-Status.

3.1. Erstellen eines Clusters

Gehen Sie folgendermaßen vor, um einen laufenden Cluster zu erstellen:

1. Authentifizieren Sie die Knoten, aus denen der Cluster gebildet werden soll.
2. Konfigurieren und synchronisieren Sie die Cluster-Knoten.
3. Starten Sie die Cluster-Dienste auf den Cluster-Knoten.

Die folgenden Abschnitte beschreiben die Befehle, die Sie für diese Schritte benötigen.

3.1.1. Authentifizieren der Cluster-Knoten

Der folgende Befehl authentifiziert **pcs** beim **pcs**-Daemon auf den Knoten im Cluster.

- Der Benutzername für den **pcs**-Administrator muss auf jedem Knoten **hacluster** lauten. Es wird empfohlen, dass auf jedem Knoten dasselbe Passwort für den **hacluster**-Benutzer verwendet wird.
- Falls Sie keinen Benutzernamen und kein Passwort angeben, fordert Sie das System für jeden Knoten zu deren Eingabe auf, wenn Sie den Befehl ausführen.
- Falls Sie keine Knoten angeben, authentifiziert dieser Befehl **pcs** auf jenen Knoten, die mit dem Befehl **pcs cluster setup** angegeben wurden, falls Sie diesen Befehl zuvor ausgeführt haben.

```
pcs cluster auth [node] [...] [-u username] [-p password]
```

Autorisierungs-Tokens werden in der Datei **~/.pcs/tokens** (oder **/var/lib/pcsd/tokens**) gespeichert.

3.1.2. Konfigurieren und Starten der Cluster-Knoten

Der folgende Befehl konfiguriert die Cluster-Konfigurationsdatei und synchronisiert die Konfiguration auf den angegebenen Knoten.

- Falls Sie die Option **--start** angeben, startet der Befehl zudem die Cluster-Dienste auf den angegebenen Knoten. Falls notwendig, können Sie die Cluster-Dienste auch mit dem separaten Befehl **pcs cluster start** starten.
- Falls Sie die Option **--local** angeben, führt der Befehl die Änderungen nur auf dem lokalen Knoten durch.

```
pcs cluster setup [--start] [--local] --name cluster_ name node1 [node2]
[...]
```

Der folgende Befehl startet die Cluster-Dienste auf einem oder mehreren angegebenen Knoten.

- Falls Sie die Option **--all** angeben, startet der Befehl die Cluster-Dienste auf allen Knoten.

- Falls Sie keine Knoten angeben, werden die Cluster-Dienste nur auf dem lokalen Knoten gestartet.

```
pcs cluster start [--all] [node] [...]
```

3.2. Verwalten von Cluster-Knoten

Die folgenden Abschnitte beschreiben die Befehle, die Sie zur Verwaltung von Cluster-Knoten verwenden, darunter Befehle zum Starten und Stoppen von Cluster-Diensten und zum Hinzufügen und Entfernen von Cluster-Knoten.

3.2.1. Stoppen von Cluster-Diensten

Der folgende Befehl stoppt Cluster-Dienste auf einem oder mehreren angegebenen Knoten. Wie auch beim Befehl **pcs cluster start** stoppt die Option **--all** die Cluster-Dienste auf allen Knoten und falls Sie keinen Knoten angeben, werden die Cluster-Dienste nur auf dem lokalen Knoten gestoppt.

```
pcs cluster stop [--all] [node] [...]
```

Sie können den Stopp der Cluster-Dienste auf dem lokalen Knoten mithilfe des folgenden Befehls erzwingen, wodurch der Befehl **kill -9** ausgeführt wird.

```
pcs cluster kill
```

3.2.2. Aktivieren und Deaktivieren der Cluster-Dienste

Verwenden Sie den folgenden Befehl, um die Cluster-Dienste auf einem oder mehreren angegebenen Knoten so zu konfigurieren, dass diese beim Systemstart starten.

- Falls Sie die Option **--all** angeben, aktiviert der Befehl die Cluster-Dienste auf allen Knoten.
- Falls Sie keine Knoten angeben, werden die Cluster-Dienste nur auf dem lokalen Knoten aktiviert.

```
pcs cluster enable [--all] [node] [...]
```

Verwenden Sie den folgenden Befehl, um die Cluster-Dienste auf einem oder mehreren angegebenen Knoten so zu konfigurieren, dass diese nicht beim Systemstart starten.

- Falls Sie die Option **--all** angeben, deaktiviert der Befehl die Cluster-Dienste auf allen Knoten.
- Falls Sie keine Knoten angeben, werden die Cluster-Dienste nur auf dem lokalen Knoten deaktiviert.

```
pcs cluster disable [--all] [node] [...]
```

3.2.3. Hinzufügen und Entfernen von Cluster-Knoten

Der folgende Befehl fügt einen neuen Knoten zu einem vorhandenen Cluster hinzu. Dieser Befehl synchronisiert zudem die Cluster-Konfigurationsdatei **cluster.conf** auf allen Knoten im Cluster, einschließlich dem neu hinzugefügten Knoten.

```
pcs cluster node add node
```

Der folgende Befehl fährt den angegebenen Knoten herunter und entfernt ihn auf allen anderen Knoten im Cluster aus der Cluster-Konfigurationsdatei **cluster.conf**. Informationen über das Entfernen sämtlicher Informationen über den Cluster von den Cluster-Knoten, um den Cluster dauerhaft zu löschen, finden Sie unter Abschnitt 3.4, „Entfernen der Cluster-Konfiguration“.

```
pcs cluster node remove node
```

3.2.4. Standby-Modus

Der folgende Befehl versetzt den angegebenen Knoten in den Standby-Modus. Der angegebene Knoten ist dadurch nicht länger in der Lage, Ressourcen zu hosten. Jegliche derzeit auf dem Knoten aktive Ressourcen werden auf einen anderen Knoten verlegt. Falls Sie die Option **--all** angeben, versetzt dieser Befehl alle Knoten in den Standby-Modus.

Sie können diesen Befehl bei der Aktualisierung von Paketen einer Ressource verwenden. Der Befehl kann auch zum Testen einer Konfiguration verwendet werden, um die Wiederherstellung zu testen, ohne tatsächlich den Knoten herunterzufahren.

```
pcs cluster standby node | --all
```

Der folgende Befehl hebt den Standby-Modus für den angegebenen Knoten auf. Nach Ausführen dieses Befehls ist der angegebene Knoten wieder in der Lage, Ressourcen zu hosten. Falls Sie die Option **--all** angeben, hebt dieser Befehl den Standby-Modus für alle Knoten auf.

```
pcs cluster unstandby node | --all
```

Beachten Sie, dass die Ausführung des Befehls **pcs cluster standby** den Ressourcen eine Beschränkung auferlegt, um diese daran zu hindern, auf dem angegebenen Knoten zu laufen. Wenn Sie den Befehl **pcs cluster unstandby** ausführen, wird diese Beschränkung entfernt. Dadurch werden die Ressourcen nicht automatisch auf den angegebenen Knoten zurückverlegt; wo die Ressourcen zu diesem Zeitpunkt ausgeführt werden können, hängt von der ursprünglichen Konfiguration Ihrer Ressourcen ab. Informationen über Ressourcenbeschränkungen finden Sie in Kapitel 6, *Ressourcenbeschränkungen*.

3.3. Festlegen von Benutzerberechtigungen

Ab Red Hat Enteprise Linux 6.6 können Sie den Befehl **pcs acl** dazu verwenden, um mittels Zugriffssteuerungslisten (engl.: Access Control Lists oder ACLs) Berechtigungen für lokale Benutzer einzustellen, die Leseberechtigungen oder Lese- und Schreibberechtigungen für die Cluster-Konfiguration gewähren.

Das Einstellen der Berechtigungen für lokale Benutzer erfordert zwei Schritte:

1. Führen Sie den Befehl **pcs acl role create...** aus, um eine *Rolle* zu erstellen, welche die Berechtigungen für diese Rolle definiert.

2. Diese von Ihnen erstellte Rolle können Sie mithilfe des Befehls **pcs acl user create** einem Benutzer zuweisen.

Das folgende Beispielverfahren gewährt einem lokalen Benutzer namens **rouser** Lesezugriff auf eine Cluster-Konfiguration.

1. Dieses Verfahren erfordert, dass der Benutzer **rouser** auf dem lokalen System existiert und dass der Benutzer **rouser** Mitglied der Gruppe **haclu ster** ist.

```
# adduser rouser
# usermod -a -G hacluster rouser
```

2. Aktivieren Sie Pacemaker-ACLs mit der Cluster-Eigenschaft **`enable-acl`**.

```
# pcs property set enable-acl=true --force
```

3. Erstellen Sie eine Rolle namens **`read-only`** mit Leseberechtigungen für die CIB.

```
# pcs acl role create read-only description="Read access to
cluster" read xpath /cib
```

4. Erstellen Sie den Benutzer **`rouser`** im ACL-System von pcs und weisen Sie diesem Benutzer die Rolle **`read-only`** zu.

```
# pcs acl user create rouser read-only
```

5. Zeigen Sie alle derzeitigen ACLs an.

```
# pcs acl
User: rouser
  Roles: read-only
Role: read-only
  Description: Read access to cluster
  Permission: read xpath /cib (read-only-read)
```

Das folgende Beispielverfahren gewährt einem lokalen Benutzer namens **`wuser`** Schreibzugriff auf eine Cluster-Konfiguration.

1. Dieses Verfahren erfordert, dass der Benutzer **`wuser`** auf dem lokalen System existiert und dass der Benutzer **`wuser`** Mitglied der Gruppe **`hacluster`** ist.

```
# adduser wuser
# usermod -a -G hacluster wuser
```

2. Aktivieren Sie Pacemaker-ACLs mit der Cluster-Eigenschaft **`enable-acl`**.

```
# pcs property set enable-acl=true --force
```

3. Erstellen Sie eine Rolle namens **`write-access`** mit Schreibberechtigungen für die CIB.

```
# pcs acl role create write-access description="Full access" write
xpath /cib
```

4. Erstellen Sie den Benutzer **`wuser`** im ACL-System von pcs und weisen Sie diesem Benutzer die Rolle **`write-access`** zu.

```
# pcs acl user create wuser write-access
```

5. Zeigen Sie alle derzeitigen ACLs an.

```
# pcs acl
```

```
User: rouser
  Roles: read-only
User: wuser
  Roles: write-access
Role: read-only
  Description: Read access to cluster
  Permission: read xpath /cib (read-only-read)
Role: write-access
  Description: Full Access
  Permission: write xpath /cib (write-access-write)
```

Weitere Informationen über Cluster-ACLs finden Sie auf dem Hilfebildschirm des Befehls **pcs acl**.

3.4. Entfernen der Cluster-Konfiguration

Mithilfe des folgenden Befehls können Sie sämtliche Cluster-Konfigurationsdateien entfernen und alle Cluster-Dienste stoppen, wodurch der Cluster dauerhaft gelöscht wird.



Warnung

Dieser Befehl entfernt dauerhaft jegliche erstellte Cluster-Konfiguration. Es wird empfohlen, dass Sie **pcs cluster stop** ausführen, bevor Sie den Cluster löschen.

```
pcs cluster destroy
```

3.5. Anzeigen des Cluster-Status

Der folgende Befehl zeigt den derzeitigen Status des Clusters und der Cluster-Ressourcen an.

```
pcs status
```

Mithilfe der folgenden Befehle können Sie Informationen mit etwas eingeschränktem Umfang über den aktuellen Cluster-Status anzeigen.

Der folgende Befehl zeigt den Status des Clusters, nicht jedoch der Cluster-Ressourcen.

```
pcs cluster status
```

Der folgende Befehl zeigt den Status der Cluster-Ressourcen.

```
pcs status resources
```

Kapitel 4. Fencing: Konfigurieren von STONITH

STONITH ist ein Akronym für Shoot-The-Other-Node-In-The-Head und soll Ihre Daten vor der Beschädigung durch fehlerhafte Knoten oder durch zeitgleichen Zugriff schützen.

Nur weil ein Knoten nicht mehr reagiert, bedeutet dies nicht, dass er nicht mehr auf Ihre Daten zugreift. Die einzige zuverlässige Methode, um zu gewährleisten, dass Ihre Daten sicher sind, ist das Abgrenzen des Knotens mit STONITH. Erst dann ist sichergestellt, dass der Knoten offline ist, und einem anderen Knoten darf Zugriff auf die Daten gewährt werden.

STONITH spielt ebenfalls eine Rolle in Situationen, in denen ein geclusterter Dienst nicht gestoppt werden kann. In diesem Fall verwendet der Cluster STONITH, um den gesamten Knoten offline zu zwingen, woraufhin es sicher ist, den Dienst auf einem anderen Knoten zu starten.

4.1. Verfügbare STONITH (Fencing)-Agenten

Verwenden Sie den folgenden Befehl, um eine Liste aller verfügbaren STONITH-Agenten anzuzeigen. Wenn Sie einen Filter angeben, zeigt dieser Befehl nur jene STONITH-Agenten, die diesem Filter entsprechen.

```
pcs stonith list [filter]
```

4.2. Allgemeine Eigenschaften von Fencing-Geräten



Anmerkung

Um ein Fencing-Gerät zu deaktivieren, können Sie die **`target-role`** einstellen wie für eine normale Ressource auch.



Anmerkung

Um einen bestimmten Knoten daran zu hindern, ein Fencing-Gerät zu verwenden, können Sie Standortbeschränkungen verwenden.

Tabelle 4.1, „Allgemeine Eigenschaften von Fencing-Geräten“ beschreibt die allgemeinen Eigenschaften, die Sie für Fencing-Geräte einstellen können. In Abschnitt 4.3, „Anzeigen gerätespezifischer Fencing-Optionen“ finden Sie Informationen über Fencing-Eigenschaften, die Sie für bestimmte Fencing-Geräte festlegen können.



Anmerkung

Informationen über erweiterte Fencing-Konfigurationseigenschaften finden Sie in Abschnitt 4.9, „Weitere Fencing-Konfigurationsoptionen“

Tabelle 4.1. Allgemeine Eigenschaften von Fencing-Geräten

Feld	Typ	Standard	Beschreibung
stonith-timeout	Zeitspanne	60s	Wert für die Zeitüberschreitung einer STONITH-Aktion pro stonith-Gerät. Setzt die Cluster-Eigenschaft **stonith-timeout** außer Kraft
priority	Ganzzahl	0	Die Priorität der stonith-Ressource. Geräte werden in der Reihenfolge von höchster nach niedrigster Priorität versucht
pcmk_host_map	Zeichenfolge		Eine Zuordnung von Hostnamen zu Portnummern für Geräte, die keine Hostnamen unterstützen. Zum Beispiel: **node1:1;node2:2,3** weist den Cluster dazu an, Port 1 für Knoten 1 und Ports 2 und 3 für Knoten 2 zu verwenden
pcmk_host_list	Zeichenfolge		Eine Liste aller Rechner, die von diesem Gerät gesteuert werden. (Optional, es sei denn **pcmk_host_check=static-list**)
pcmk_host_check	Zeichenfolge	dynamic-list	Wie bestimmt werden soll, welche Rechner von dem Gerät gesteuert werden. Zulässige Werte: **dynamic-list** (fragt das Gerät ab), **static-list** (prüft das Attribut **pcmk_host_list**), kein Wert (nimmt an, dass jedes Gerät jeden Rechner abgrenzen kann)

4.3. Anzeigen gerätespezifischer Fencing-Optionen

Verwenden Sie den folgenden Befehl, um die Optionen für den angegebenen STONITH-Agenten anzuzeigen.

```
pcs stonith describe stonith_agent
```

Beispielsweise zeigt der folgende Befehl die Optionen für den Fencing-Agenten für APC über Telnet/SSH.

```
# pcs stonith describe fence_apc
Stonith options for: fence_apc
  ipaddr (required): IP Address or Hostname
  login (required): Login Name
  passwd: Login password or passphrase
  passwd_script: Script to retrieve password
  cmd_prompt: Force command prompt
  secure: SSH connection
  port (required): Physical plug number or name of virtual machine
  identity_file: Identity file for ssh
  switch: Physical switch number on device
  inet4_only: Forces agent to use IPv4 addresses only
  inet6_only: Forces agent to use IPv6 addresses only
  ipport: TCP port to use for connection with device
  action (required): Fencing Action
```

```
  verbose: Verbose mode
  debug: Write debug information to given file
  version: Display version information and exit
  help: Display help and exit
  separator: Separator for CSV created by operation list
  power_timeout: Test X seconds for status change after ON/OFF
  shell_timeout: Wait X seconds for cmd prompt after issuing command
  login_timeout: Wait X seconds for cmd prompt after login
  power_wait: Wait X seconds after issuing ON/OFF
  delay: Wait X seconds before fencing is started
  retry_on: Count of attempts to retry power on
```

4.4. Erstellen eines Fencing-Geräts

Der folgende Befehl erstellt ein stonith-Gerät.

```
pcs stonith create stonith_id stonith_device_type
[stonith_device_options]
```

```
# pcs stonith create MyStonith fence_virt pcmk_host_list=f1 op monitor
interval=30s
```

Wenn Sie ein einziges Fencing-Gerät für mehrere Knoten nutzen und einen anderen Port für jeden Knoten verwenden, dann brauchen Sie nicht für jeden Knoten ein separates Gerät zu erstellen. Sie können stattdessen die Option **pcmk_host_map** verwenden, um zu definieren, welcher Port zu welchem Knoten gehört. Beispielsweise erstellt der folgende Befehl ein einziges Fencing-Gerät namens **myapc-west-13**, das einen APC-Powerswitch namens **west-apc** verwendet und Port 15 für Knoten **west-13** nutzt.

```
# pcs stonith create myapc-west-13 fence_apc pcmk_host_list="west-13"
ipaddr="west-apc" login="apc" passwd="apc" port="15"
```

Das folgende Beispiel dagegen verwendet den APC-Powerswitch namens **west-apc** zum Abgrenzen von Knoten **west-13** auf Port 15, **west-14** auf Port 17, **west-15** auf Port 18 und **west-16** auf Port 19.

```
# pcs stonith create myapc fence_apc pcmk_host_list="west-13,west-
14,west-15,west-16" pcmk_host_map="west-13:15;west-14:17;west-
15:18;west-16:19" ipaddr="west-apc" login="apc" passwd="apc"
```

4.5. Konfigurieren von Storage-basierten Fencing-Geräten mit Aufheben der Abgrenzung

Bei der Erstellung eines SAN/Storage-Fencing-Geräts (also ein Fencing-Gerät mit einem Fencing-Agent, der nicht auf der Stromversorgung basiert), müssen Sie die Metaoption **provides=unfencing** beim Erstellen des **stonith**-Geräts angeben. Dadurch wird sichergestellt, dass die Abgrenzung eines abgegrenzten Knotens wieder aufgehoben wird, bevor der Knoten neu gestartet wird und die Cluster-Dienste auf diesem Knoten gestartet werden.

Das Angeben der Metaoption **provides=unfencing** ist bei der Konfiguration eines strombasierten Fencing-Geräts nicht notwendig, da das Gerät selbst die Stromversorgung für den Knoten stellt, damit dieser booten und einen Wiederbeitritt zum Cluster versuchen kann. In diesem Fall impliziert

der Neustart, dass die Aufhebung der Abgrenzung erfolgt ist.

Der folgende Befehl konfiguriert ein stonith-Gerät namens **`my-scsi-shooter`**, das den Fencing-Agent **`fence_scsi`** verwendet und die Aufhebung der Abgrenzung für das Gerät ermöglicht.

```
pcs stonith create my-scsi-shooter fence_scsi devices=/dev/sda meta
provides=unfencing
```

4.6. Anzeigen von Fencing-Geräten

Der folgende Befehl zeigt alle derzeit konfigurierten Fencing-Geräte. Falls eine *stonith_id* angegeben wird, zeigt der Befehl nur die Optionen für das konfigurierte stonith-Gerät. Falls die Option **`--full`** angegeben ist, werden alle konfigurierten stonith-Optionen angezeigt.

```
pcs stonith show [stonith_id] [--full]
```

4.7. Bearbeiten und Löschen von Fencing-Geräten

Verwenden Sie den folgenden Befehl, um Optionen zu einem derzeit konfigurierten Fencing-Gerät hinzuzufügen oder zu bearbeiten.

```
pcs stonith update stonith_id [stonith_device_options]
```

Verwenden Sie den folgenden Befehl, um ein Fencing-Gerät aus der derzeitigen Konfiguration zu entfernen.

```
pcs stonith delete stonith_id
```

4.8. Verwalten von Knoten mit Fencing-Geräten

Sie können einen Knoten mithilfe des folgenden Befehls manuell abgrenzen. Falls Sie die Option **`--off`** angeben, wird der API-Aufruf **`off`** an stonith verwendet, um den Knoten abzuschalten statt ihn neu zu starten.

```
pcs stonith fence node [--off]
```

Mithilfe des folgenden Befehls können Sie prüfen, ob ein angegebener Knoten derzeit angeschaltet ist.



Warnung

Falls der angegebene Knoten noch die Cluster-Software oder -Dienste ausführt, die normalerweise vom Cluster gesteuert werden, hat dies eine Beschädigung der Daten und einen Ausfall des Clusters zur Folge.

```
pcs stonith confirm node
```

4.9. Weitere Fencing-Konfigurationsoptionen

4.9. Weitere Fencing-Konfigurationsoptionen

Tabelle 4.2, „Fortgeschrittene Eigenschaften von Fencing-Geräten“ gibt einen Überblick über weitere Eigenschaften, die Sie für Fencing-Geräte einstellen können. Beachten Sie, dass diese Eigenschaften nur für eine fortgeschrittene Verwendung nötig sind.

Tabelle 4.2. Fortgeschrittene Eigenschaften von Fencing-Geräten

Feld	Typ	Standard	Beschreibung
pcmk_host_argument	Zeichenfolge	Port	Ein alternativer Parameter, der anstelle des Ports angegeben werden soll. Einige Geräte unterstützen den standardmäßigen port-Parameter nicht oder bieten zusätzliche Parameter. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Parameters, der den abzugrenzenden Rechner angibt. Der Wert **none** kann verwendet werden, um den Cluster anzuweisen, keine weiteren Parameter zu übergeben.
pcmk_reboot_action	Zeichenfolge	reboot	Ein alternativer Befehl, der anstelle von **reboot** ausgeführt werden soll. Einige Geräte unterstützen die standardmäßigen Befehle nicht oder bieten zusätzliche Befehle. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Befehls für die „reboot“-Aktion.
pcmk_reboot_timeout	Zeitspanne	60s	Ein alternativer Wert für die Zeitüberschreitung bei Neustarts, der anstelle von **stonith-timeout** verwendet werden soll. Einige Geräte benötigen sehr viel weniger bzw. mehr Zeit für diese Aktion als normal. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Werts für die Zeitüberschreitung bei „reboot“-Aktionen.
pcmk_reboot_retries	Ganzzahl	2	Die maximale Anzahl von Neuversuchen für den **reboot**-Befehl innerhalb der Zeitspanne für die Zeitüberschreitung. Einige Geräte unterstützen keine mehrfachen Verbindungen. Operationen können fehlschlagen, falls das Gerät mit einer anderen Operation beschäftigt ist, weshalb Pacemaker die Operation automatisch erneut versucht, falls noch genügend Zeit bleibt. Verwenden Sie diese Option, um die Anzahl der Neuversuche von Pacemaker für „reboot“-Aktionen zu verändern, bevor aufgegeben werden soll.

Feld	Typ	Standard	Beschreibung
pcmk_off_action	Zeichenfolge	off	Ein alternativer Befehl, der anstelle von **off** ausgeführt werden soll. Einige Geräte unterstützen die standardmäßigen Befehle nicht oder bieten zusätzliche Befehle. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Befehls für die „off“-Aktion.
pcmk_off_timeout	Zeitspanne	60s	Ein alternativer Wert für die Zeitüberschreitung bei „off“-Aktionen, der anstelle von **stonith-timeout** verwendet werden soll. Einige Geräte benötigen sehr viel weniger bzw. mehr Zeit für diese Aktion als normal. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Werts für die Zeitüberschreitung bei „off“-Aktionen.
pcmk_off_retries	Ganzzahl	2	Die maximale Anzahl von Neuversuchen für den **off**-Befehl innerhalb der Zeitspanne für die Zeitüberschreitung. Einige Geräte unterstützen keine mehrfachen Verbindungen. Operationen können fehlschlagen, falls das Gerät mit einer anderen Operation beschäftigt ist, weshalb Pacemaker die Operation automatisch erneut versucht, falls noch genügend Zeit bleibt. Verwenden Sie diese Option, um die Anzahl der Neuversuche von Pacemaker für „off“-Aktionen zu verändern, bevor aufgegeben werden soll.
pcmk_list_action	Zeichenfolge	list	Ein alternativer Befehl, der anstelle von **list** ausgeführt werden soll. Einige Geräte unterstützen die standardmäßigen Befehle nicht oder bieten zusätzliche Befehle. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Befehls für die „list“-Aktion.
pcmk_list_timeout	Zeitspanne	60s	Ein alternativer Wert für die Zeitüberschreitung bei „list“-Aktionen, der anstelle von **stonith-timeout** verwendet werden soll. Einige Geräte benötigen sehr viel weniger bzw. mehr Zeit für diese Aktion als normal. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Werts für die Zeitüberschreitung bei „list“-Aktionen.

Feld	Typ	Standard	Beschreibung
pcmk_list_retries	Ganzzahl	2	Die maximale Anzahl von Neuversuchen für den **list**-Befehl innerhalb der Zeitspanne für die Zeitüberschreitung. Einige Geräte unterstützen keine mehrfachen Verbindungen. Operationen können fehlschlagen, falls das Gerät mit einer anderen Operation beschäftigt ist, weshalb Pacemaker die Operation automatisch erneut versucht, falls noch genügend Zeit bleibt. Verwenden Sie diese Option, um die Anzahl der Neuversuche von Pacemaker für „list“-Aktionen zu verändern, bevor aufgegeben werden soll.
pcmk_monitor_action	Zeichenfolge	monitor	Ein alternativer Befehl, der anstelle von **monitor** ausgeführt werden soll. Einige Geräte unterstützen die standardmäßigen Befehle nicht oder bieten zusätzliche Befehle. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Befehls für die „monitor“-Aktion.
pcmk_monitor_timeout	Zeitspanne	60s	Ein alternativer Wert für die Zeitüberschreitung bei „monitor“-Aktionen, der anstelle von **stonith-timeout** verwendet werden soll. Einige Geräte benötigen sehr viel weniger bzw. mehr Zeit für diese Aktion als normal. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Werts für die Zeitüberschreitung bei „monitor“-Aktionen.
pcmk_monitor_retries	Ganzzahl	2	Die maximale Anzahl von Neuversuchen für den **monitor**-Befehl innerhalb der Zeitspanne für die Zeitüberschreitung. Einige Geräte unterstützen keine mehrfachen Verbindungen. Operationen können fehlschlagen, falls das Gerät mit einer anderen Operation beschäftigt ist, weshalb Pacemaker die Operation automatisch erneut versucht, falls noch genügend Zeit bleibt. Verwenden Sie diese Option, um die Anzahl der Neuversuche von Pacemaker für „monitor“-Aktionen zu verändern, bevor aufgegeben werden soll.
pcmk_status_action	Zeichenfolge	status	Ein alternativer Befehl, der anstelle von **status** ausgeführt werden soll. Einige Geräte unterstützen die standardmäßigen Befehle nicht oder bieten zusätzliche Befehle. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Befehls für die „status“-Aktion.

Feld	Typ	Standard	Beschreibung
pcmk_status_timeout	Zeitspanne	60s	Ein alternativer Wert für die Zeitüberschreitung bei „status“-Aktionen, der anstelle von **stonith-timeout** verwendet werden soll. Einige Geräte benötigen sehr viel weniger bzw. mehr Zeit für diese Aktion als normal. Verwenden Sie dies zur Angabe eines alternativen, gerätespezifischen Werts für die Zeitüberschreitung bei „status“-Aktionen.
pcmk_status_retries	Ganzzahl	2	Die maximale Anzahl von Neuversuchen für den **status**-Befehl innerhalb der Zeitspanne für die Zeitüberschreitung. Einige Geräte unterstützen keine mehrfachen Verbindungen. Operationen können fehlschlagen, falls das Gerät mit einer anderen Operation beschäftigt ist, weshalb Pacemaker die Operation automatisch erneut versucht, falls noch genügend Zeit bleibt. Verwenden Sie diese Option, um die Anzahl der Neuversuche von Pacemaker für „status“-Aktionen zu verändern, bevor aufgegeben werden soll.

4.10. Konfigurieren von Fencing-Levels

Pacemaker unterstützt die Abgrenzung von Knoten unter Verwendung mehrerer Geräte mithilfe eines Features namens Fencing-Topologien. Um Topologien zu implementieren, erstellen Sie einzelne Geräte auf normalem Weg und definieren Sie anschließend einen oder mehrere Fencing-Levels im Konfigurationsabschnitt für Fencing-Topologien.

- Die Levels werden nacheinander in aufsteigender Reihenfolge probiert, beginnend bei 1.
- Falls ein Gerät ausfällt, wird die Verarbeitung für den derzeitigen Level beendet. In diesem Level werden keine weiteren Geräte versucht, stattdessen wird mit dem nächsten Level fortgefahren.
- Falls alle Geräte erfolgreich abgegrenzt werden, dann war dieser Level erfolgreich und keine anderen Levels werden probiert.
- Die Operation ist abgeschlossen, wenn ein Level erfolgreich war oder wenn alle Levels erfolglos versucht wurden.

Verwenden Sie den folgenden Befehl, um ein Fencing-Level zu einem Knoten hinzuzufügen. Die Geräte werden als kommagetrennte Liste mit stonith-IDs angegeben, die für diesen Knoten auf diesem Level versucht werden.

```
pcs stonith level add level node devices
```

Der folgende Befehl listet alle Fencing-Levels auf, die derzeit konfiguriert sind.

```
pcs stonith level
```

In dem folgenden Beispiel sind für den Knoten **rh7-2** zwei Fencing-Geräte konfiguriert: ein ilo-

Fencing-Gerät namens **my_ilo** und ein apc-Fencing-Gerät namens **my_apc**. Diese Befehle richten Fencing-Levels ein, sodass im Falle eines Ausfalls des Geräts **my_ilo** Pacemaker stattdessen das Gerät **my_apc** zu verwenden versuchen wird. Dieses Beispiel zeigt auch die Ausgabe des Befehls **pcs stonith level**, nachdem die Levels konfiguriert wurden.

```
# pcs stonith level add 1 rh7-2 my_ilo
# pcs stonith level add 2 rh7-2 my_apc
# pcs stonith level
 Node: rh7-2
  Level 1 - my_ilo
  Level 2 - my_apc
```

Der folgende Befehl entfernt das Fencing-Level für den angegebenen Knoten und die angegebenen Geräte. Falls keine Knoten oder Geräte angegeben werden, wird das Fencing-Level von allen Knoten entfernt.

```
pcs stonith level remove level [node_id] [stonith_id] ... [stonith_id]
```

Der folgende Befehl löscht die Fencing-Levels auf dem angegebenen Knoten oder der angegebenen stonith-ID. Falls Sie keine Knoten oder stonith-IDs angeben, werden alle Fencing-Levels gelöscht.

```
pcs stonith level clear [node|stonith_id(s)]
```

Falls Sie mehr als eine stonith-ID angeben, müssen diese durch Kommas getrennt sein und dürfen keine Leerzeichen enthalten, wie im folgenden Beispiel veranschaulicht.

```
# pcs stonith level clear dev_a,dev_b
```

Der folgende Befehl überprüft, ob alle Fencing-Geräte und Knoten, die in Fencing-Levels angegeben wurden, existieren.

```
pcs stonith level verify
```

Kapitel 5. Konfigurieren von Cluster-Ressourcen

Dieses Kapitel liefert Informationen über die Konfiguration von Ressourcen in einem Cluster.

5.1. Ressourcenerstellung

Verwenden Sie den folgenden Befehl, um eine Cluster-Ressource zu erstellen.

```
pcs resource create resource_id standard:provider:type|type [resource
options]
```

Beispielsweise erstellt der folgende Befehl eine Ressource namens VirtualIP mit dem Standard **ocf**, Provider **heartbeat** und Typ **IPaddr2**. Die Floating-IP-Adresse dieser Ressource ist 192.168.0.120 und das System prüft alle 30 Sekunden, ob die Ressource läuft.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.120
cidr_netmask=24 op monitor interval=30s
```

Alternativ können Sie die Felder *standard* und *provider* weglassen und den folgenden Befehl verwenden. Dadurch werden die Standardwerte **ocf** und **heartbeat** verwendet.

```
# pcs resource create VirtualIP IPaddr2 ip=192.168.0.120
cidr_netmask=24 op monitor interval=30s
```

Verwenden Sie den folgenden Befehl, um eine konfigurierte Ressource zu löschen.

```
pcs resource delete resource_id
```

Beispielsweise löscht der folgende Befehl eine vorhandene Ressource mit der Ressourcen-ID **VirtualIP**.

```
# pcs resource delete VirtualIP
```

- Informationen über die Felder *resource_id*, *standard*, *provider* und *type* im Befehl **pcs resource create** finden Sie in Abschnitt 5.2, „Ressourceneigenschaften“.
- Informationen über das Definieren von Ressourcenparametern für einzelne Ressourcen finden Sie in Abschnitt 5.3, „Ressourcenspezifische Parameter“.
- Informationen über das Definieren von Ressourcen-Metaoptionen, die der Cluster verwendet, um über das Verhalten der Ressource zu entscheiden, finden Sie in Abschnitt 5.4, „Ressourcen-Metaoptionen“.
- Informationen über das Definieren der Operationen, die auf einer Ressource auszuführen sind, finden Sie in Abschnitt 5.5, „Ressourcenoperationen“.

5.2. Ressourceneigenschaften

Die Eigenschaften, die Sie für eine Ressource definieren, teilen dem Cluster mit, welches Skript für die Ressource zu verwenden ist, wo dieses Skript zu finden ist und welchen Standards es entspricht. Tabelle 5.1, „Ressourceneigenschaften“ beschreibt diese Eigenschaften.

Tabelle 5.1. Ressourceneigenschaften

Feld	Beschreibung
resource_id	Ihr Name für die Ressource
standard	Der Standard, dem das Skript entspricht. Zulässige Werte: **`ocf`**, **`service`**, **`upstart`**, **`systemd`**, **`lsb`**, **`stonith`**
type	Der Name des Ressourcen-Agents, den Sie verwenden möchten, z. B. **`IPaddr`** oder **`Filesystem`**
provider	Die OCF-Spezifikationen erlauben es mehreren Anbietern, denselben Ressourcen-Agent bereitzustellen. Die meisten der von Red Hat bereitgestellten Agents verwenden **`heartbeat`** als Provider.

Tabelle 5.2, „Befehle zum Anzeigen von Ressourceneigenschaften“ zeigt eine Übersicht über die Befehle, welche die verfügbaren Ressourceneigenschaften anzeigen.

Tabelle 5.2. Befehle zum Anzeigen von Ressourceneigenschaften

pcs-Anzeigebefehl	Ausgabe
`pcs resource list`	Zeigt eine Liste aller verfügbaren Ressourcen
`pcs resource standard`	Zeigt eine Liste der verfügbaren Standards für Ressourcen-Agents
`pcs resource providers`	Zeigt eine Liste der verfügbaren Provider für Ressourcen-Agents
`pcs resource list` ***`string`***	Zeigt eine Liste verfügbarer Ressourcen, gefiltert nach der angegebenen Zeichenfolge. Sie können mit diesem Befehl Ressourcen gefiltert nach dem Namen eines Standards, Providers oder Typs anzeigen

5.3. Ressourcenspezifische Parameter

Für eine einzelne Ressource können Sie den folgenden Befehl verwenden, um die Parameter anzuzeigen, die Sie für diese Ressource festlegen können.

```
# pcs resource describe standard:provider:type|type
```

Beispielsweise zeigt der folgende Befehl die Parameter, die Sie für eine Ressource des Typs **`LVM`** festlegen können.

```
# pcs resource describe LVM
Resource options for: LVM
  volgrpname (required): The name of volume group.
  exclusive: If set, the volume group will be activated exclusively.
  partial_activation: If set, the volume group will be activated even
  only partial of the physicalvolumes available. It helps to set to
  true, when you are using mirroring logical volumes.
```

5.4. Ressourcen-Metaoptionen

Zusätzlich zu den ressourcenspezifischen Parametern können Sie weitere Ressourcenoptionen für jede Ressource festlegen. Diese Optionen werden vom Cluster verwendet, um über das Verhalten Ihrer Ressource zu entscheiden. Tabelle 5.3, „Ressourcen-Metaoptionen“ beschreibt diese Optionen.

Tabelle 5.3. Ressourcen-Metaoptionen

Feld	Standard	Beschreibung
`priority`	**`0`**	Falls nicht alle Ressourcen aktiv sein können, wird der Cluster Ressourcen mit niedriger Priorität stoppen, um jene mit hoher Priorität weiterhin aktiv ausführen zu können.
`target-role`	**`Started`**	In welchem Status soll der Cluster die Ressource zu halten versuchen? Zulässige Werte: * *Stopped* – Zwingt die Ressource in einen gestoppten Status * *Started* – Erlaubt einen Start der Ressource (im Falle von Multi-Status-Ressourcen werden diese nicht in den Master-Status hochgestuft) * *Master* – Erlaubt einen Start der Ressource sowie ggf. das Hochstufen
`is-managed`	**`true`**	Darf der Cluster die Ressource stoppen und starten? Zulässige Werte: **`true`**, **`false`**
`resource-stickiness`	0	Wert, der anzeigt, wie stark die Ressource an ihrem derzeitigen Standort bleiben möchte.

Feld	Standard	Beschreibung
requires	Berechnet	Legt fest, unter welchen Umständen die Ressource gestartet werden kann. Standardmäßig **fencing**, ausgenommen in den unten genannten Situationen. Mögliche Werte: * **nothing** – Der Cluster kann die Ressource immer starten. * **quorum** – Der Cluster kann diese Ressource nur starten, wenn die Mehrzahl der konfigurierten Knoten aktiv ist. Dies ist der Standardwert, wenn **stonith-enabled** den Wert **false** hat oder falls der Standardwert **standard** der Ressource **stonith** lautet. * **fencing** – Der Cluster kann diese Ressource nur starten, wenn die Mehrzahl der konfigurierten Knoten aktiv ist *und* wenn jegliche ausgefallene oder unbekannte Knoten abgeschaltet wurden. * **unfencing** – Der Cluster kann diese Ressource nur starten, wenn die Mehrzahl der konfigurierten Knoten aktiv ist *und* wenn jegliche ausgefallene oder unbekannte Knoten abgeschaltet wurden *und* nur auf Knoten, deren *Abgrenzung aufgehoben* wurde. Dies ist der Standardwert, falls die **stonith**-Metaoption **provides=unfencing** für ein Fencing-Gerät festgelegt wurde. Informationen über die **stonith**-Metaoption **provides=unfencing** finden Sie in Abschnitt 4.5, „Konfigurieren von Storage-basierten Fencing-Geräten mit Aufheben der Abgrenzung“.
migration-threshold	**INFINITY** (deaktiviert)	Wie viele Ausfälle dürfen für diese Ressource auf einem Knoten auftreten, bevor dieser Knoten als ungeeignet zum Hosten dieser Ressource gekennzeichnet wird. Informationen über die Konfiguration der Option **migration-threshold** finden Sie in Abschnitt 7.2, „Verlegen von Ressourcen wegen Ausfall“.
failure-timeout	**0** (deaktiviert)	Verwendet zusammen mit der Option **migration-threshold**; zeigt an, wie viele Sekunden gewartet wird, bevor fortgefahren wird, als sei der Ausfall nicht geschehen, wodurch die Ressource möglicherweise wieder auf dem Knoten erlaubt wird, auf dem sie ausgefallen ist. Informationen über die Konfiguration der Option **failure-timeout** finden Sie in Abschnitt 7.2, „Verlegen von Ressourcen wegen Ausfall“.

Feld	Standard	Beschreibung
multiple-active	**stop_start**	Was soll der Cluster tun, falls er feststellt, dass die Ressource auf mehr als einem Knoten aktiv ist? Zulässige Werte: * **block** – Ressource als nicht verwaltet markieren * **stop_only** – alle aktiven Instanzen stoppen und sie so belassen * **stop_start** – alle aktiven Instanzen stoppen und die Ressource an nur einem Standort starten

Verwenden Sie den folgenden Befehl, um den Standardwert einer Ressourcenoption zu ändern.

```
pcs resource defaults options
```

Beispielsweise setzt der folgende Befehl den Standardwert für **resource-stickiness** auf 100.

```
# pcs resource defaults resource-stickiness=100
```

Wenn Sie den Parameter *options* vom Befehl **pcs resource defaults** weglassen, zeigt dieser eine Liste aller derzeit konfigurierten Standardwerte für Ressourcenoptionen an. Das folgende Beispiel zeigt die Ausgabe dieses Befehls, nachdem Sie den Standardwert für **resource-stickiness** auf 100 geändert haben.

```
# pcs resource defaults
resource-stickiness:100
```

Ungeachtet dessen, ob Sie den Standardwert einer Ressourcen-Metaoption verändert haben oder nicht, können Sie eine Ressourcenoption für eine bestimmte Ressource während der Ressourcenerstellung auf einen anderen Wert als den Standardwert festlegen. Nachfolgend sehen Sie das Format des Befehls **pcs resource create**, den Sie zur Angabe eines Werts für eine Ressourcen-Metaoption verwenden.

```
pcs resource create resource_id standard:provider:type|type [resource
options] [meta meta_options...]
```

Beispielsweise erstellt der folgende Befehl eine Ressource mit einem Wert von 50 für **resource-stickiness**.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.120
cidr_netmask=24 meta resource-stickiness=50
```

Mit dem folgenden Befehl können Sie auch den Wert einer Ressourcen-Metaoption für eine vorhandene Ressource, Gruppe, geklonte Ressource oder Master-Ressource festlegen.

```
pcs resource meta resource_id | group_id | clone_id | master_id
meta_options
```

In dem folgenden Beispiel gibt es eine vorhandene Ressource namens **dummy_resource**. Dieser Befehl setzt die Metaoption **failure-timeout** auf 20 Sekunden, sodass die Ressource nach 20 Sekunden einen Neustart auf derselben Ressource versuchen kann.

```
# pcs resource meta dummy_resource failure-timeout=20s
```

Nach Ausführen dieses Befehls können Sie die Werte für die Ressource anzeigen lassen, um zu überprüfen, ob **failure-timeout=20s** eingestellt wurde.

```
# pcs resource show dummy_resource
 Resource: dummy_resource (class=ocf provider=heartbeat type=Dummy)
  Meta Attrs: failure-timeout=20s
  Operations: start interval=0s timeout=20 (dummy_resource-start-timeout-
20)
              stop interval=0s timeout=20 (dummy_resource-stop-timeout-
20)
              monitor interval=10 timeout=20 (dummy_resource-monitor-
interval-10)
```

Informationen über Ressourcenklon-Metaoptionen finden Sie in Abschnitt 8.1, „Ressourcen-Klone“. Informationen über Ressourcen-Master-Metaoptionen finden Sie in Abschnitt 8.2, „Multi-Status-Ressourcen: Ressourcen mit mehreren Modi“.

5.5. Ressourcenoperationen

Um sicherzugehen, dass Ressourcen funktionstüchtig bleiben, können Sie eine Überwachungsoperation zur Ressourcendefinition hinzufügen. Falls Sie keine Überwachungsoperation für eine Ressource angeben, wird der **pcs**-Befehl standardmäßig eine Überwachungsoperation erstellen mit einem Intervall, das vom Ressourcen-Agent bestimmt wird. Falls der Ressourcen-Agent kein standardmäßiges Überwachungsintervall vorgibt, erstellt der pcs-Befehl eine Überwachungsoperation mit einem Intervall von 60 Sekunden.

Tabelle 5.4, „Eigenschaften einer Operation“ zeigt eine Übersicht über die Eigenschaften einer Operation zur Ressourcenüberwachung.

Tabelle 5.4. Eigenschaften einer Operation

Feld	Beschreibung
id	Eindeutiger Name für die Aktion. Das System weist diesen zu, wenn Sie eine Operation konfigurieren.
name	Die auszuführende Aktion. Übliche Werte: **monitor**, **start**, **stop**
interval	In welchen Abständen (in Sekunden) die Operation ausgeführt werden soll. Standardwert: **0**, also nie.
timeout	Wie lange gewartet werden soll, bevor die Aktion als fehlgeschlagen gilt. Falls Sie feststellen, dass Ihr System eine Ressource enthält, die eine lange Zeit zum Starten oder Stoppen oder zur Durchführung einer sich nicht wiederholenden Überwachungsaktion beim Start benötigt, und dass es mehr Zeit als vom System erlaubt erfordert, bevor die Startaktion als gescheitert gilt, können Sie diesen Wert vom Standardwert 20 erhöhen oder den Wert für **timeout** in „op defaults“ ändern.

Feld	Beschreibung
on-fail	Die durchzuführende Aktion, falls diese Aktion fehlschlägt. Zulässige Werte: * **ignore** – Fortfahren, als sei die Ressource nicht ausgefallen * **block** – Keine weiteren Operationen auf der Ressource ausführen * **stop** – Ressource stoppen und nicht an anderer Stelle starten * **restart** – Ressource stoppen und neu starten (möglicherweise auf einem anderen Knoten) * **fence** – Knoten, auf dem die Ressource ausgefallen ist, mit STONITH abgrenzen * **standby** – *Alle* Ressourcen von dem Knoten, auf der die Ressource ausgefallen ist, wegverlegen Die **stop**-Operation lautet **fence**, wenn STONITH aktiviert ist, andernfalls lautet sie **block**. Alle anderen Operationen lauten standardmäßig **restart**.
enabled	Falls **false**, wird die Operation so behandelt, als existierte sie nicht. Zulässige Werte: **true**, **false**

Mit dem folgenden Befehl können Sie bei der Erstellung einer Ressource Überwachungsoperationen konfigurieren.

```
pcs resource create resource_id standard:provider:type|type
[resource_options] [op operation_action operation_options [operation_type
operation_options]...]
```

Beispielsweise erstellt der folgende Befehl eine **IPaddr2**-Ressource mit einer Überwachungsoperation. Die neue Ressource trägt den Namen **VirtualIP**, hat die IP-Adresse 192.168.0.99 und die Netzmaske 24 auf **eth2**. Eine Überwachungsoperation wird alle 30 Sekunden durchgeführt.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.99
cidr_netmask=24 nic=eth2 op monitor interval=30s
```

```
# pcs resource create my_address IPaddr2 ip=10.20.30.40
cidr_netmask=24 op monitor
```

Alternativ können Sie mit dem folgenden Befehl eine Überwachungsoperation zu einer vorhandenen Ressource hinzufügen.

```
pcs resource op add resource_id operation_action [operation_properties]
```

Verwenden Sie den folgenden Befehl, um eine konfigurierte Ressourcenoperation zu löschen.

```
pcs resource op remove resource_id operation_name operation_properties
```



Anmerkung

Sie müssen die exakten Operationseigenschaften angeben, um eine vorhandene Operation richtig zu entfernen.

Um die Werte einer Überwachungsoption zu ändern, entfernen Sie die vorhandene Operation und fügen Sie anschließend die neue Operation hinzu. Beispielsweise können Sie eine **VirtualIP** mit dem folgenden Befehl erstellen.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.99
cidr_netmask=24 nic=eth2
```

Standardmäßig erstellt dieser Befehl diese Operationen.

```
Operations: start interval=0s timeout=20s (VirtualIP-start-timeout-20s)
            stop interval=0s timeout=20s (VirtualIP-stop-timeout-20s)
            monitor interval=10s timeout=20s (VirtualIP-monitor-
interval-10s)
```

Führen Sie die folgenden Befehle aus, um die Stopp-Timeout-Operation zu ändern.

```
# pcs resource op remove VirtualIP stop interval=0s timeout=20s
# pcs resource op add VirtualIP stop interval=0s timeout=40s

# pcs resource show VirtualIP
 Resource: VirtualIP (class=ocf provider=heartbeat type=IPaddr2)
  Attributes: ip=192.168.0.99 cidr_netmask=24 nic=eth2
  Operations: start interval=0s timeout=20s (VirtualIP-start-timeout-20s)
              monitor interval=10s timeout=20s (VirtualIP-monitor-
interval-10s)
              stop interval=0s timeout=40s (VirtualIP-name-stop-interval-
0s-timeout-40s)
```

Verwenden Sie den folgenden Befehl, um globale Standardwerte für Überwachungsoperationen festzulegen.

```
pcs resource op defaults [options]
```

Beispielsweise legt der folgende Befehl einen globalen Standardwert für **timeout** von 240 Sekunden für alle Überwachungsoperationen fest.

```
# pcs resource op defaults timeout=240s
```

Um die derzeit konfigurierten Standardwerte für Überwachungsoperationen anzuzeigen, geben Sie keine Optionen an, wenn Sie den Befehl **pcs resource op defaults** ausführen.

Beispielsweise zeigt der folgende Befehl die Standardwerte für Überwachungsoperationen in einem Cluster, der mit einem **timeout**-Wert von 240 Sekunden konfiguriert wurde.

```
# pcs resource op defaults
timeout: 240s
```

5.6. Anzeigen konfigurierter Ressourcen

Verwenden Sie den folgenden Befehl, um eine Liste aller konfigurierten Ressourcen anzuzeigen.

```
pcs resource show
```

Falls Ihr System beispielsweise mit einer Ressource namens **VirtualIP** und einer Ressource namens **WebSite** konfiguriert ist, dann zeigt der Befehl **pcs resource show** die folgende Ausgabe.

```
# pcs resource show
 VirtualIP (ocf::heartbeat:IPaddr2): Started
 WebSite (ocf::heartbeat:apache): Started
```

Verwenden Sie die Option **--full** des Befehls **pcs resource show** wie im folgenden Beispiel, um eine Liste aller konfigurierten Ressourcen und deren konfigurierten Parameter anzuzeigen.

```
# pcs resource show --full
 Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
  Attributes: ip=192.168.0.120 cidr_netmask=24
  Operations: monitor interval=30s
 Resource: WebSite (type=apache class=ocf provider=heartbeat)
  Attributes: statusurl=http://localhost/server-status
configfile=/etc/httpd/conf/httpd.conf
  Operations: monitor interval=1min
```

Verwenden Sie den folgenden Befehl, um die konfigurierten Parameter für eine Ressource anzuzeigen.

```
pcs resource show resource_id
```

Beispielsweise zeigt der folgende Befehl die derzeit konfigurierten Parameter der Ressource **VirtualIP** an.

```
# pcs resource show VirtualIP
 Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
  Attributes: ip=192.168.0.120 cidr_netmask=24
  Operations: monitor interval=30s
```

5.7. Ändern von Ressourcenparametern

Verwenden Sie den folgenden Befehl, um die Parameter einer konfigurierten Ressource zu ändern.

```
pcs resource update resource_id [resource_options]
```

Die folgende Befehlssequenz zeigt die ursprünglichen Werte der konfigurierten Parameter für die Ressource **VirtualIP**, die Befehle zum Ändern des Werts für den Parameter **ip** sowie die Werte nach erfolgter Änderung.

```
# pcs resource show VirtualIP
 Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
```

```
  Attributes: ip=192.168.0.120 cidr_netmask=24
  Operations: monitor interval=30s
# pcs resource update VirtualIP ip=192.169.0.120
# pcs resource show VirtualIP
 Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
  Attributes: ip=192.169.0.120 cidr_netmask=24
  Operations: monitor interval=30s
```

5.8. Mehrere Überwachungsoperationen

Sie können eine einzelne Ressource mit so vielen Überwachungsoperationen konfigurieren, wie ein Ressourcen-Agent unterstützt. Auf diese Weise können Sie jede Minute einen oberflächlichen Funktionstest durchführen sowie in größeren Abständen zunehmend intensive Funktionstests.



Anmerkung

Bei der Konfiguration von mehreren Überwachungsoperationen müssen Sie sicherstellen, dass keine zwei Operationen mit demselben Intervall durchgeführt werden.

Um weitere Überwachungsoperationen für eine Ressource hinzuzufügen, die intensivere Tests auf verschiedenen Levels unterstützt, können Sie die Option **OCF_CHECK_LEVEL=*n*** hinzufügen.

Falls Sie beispielsweise die folgende **IPaddr2**-Ressource konfigurieren, erstellt dies standardmäßig eine Überwachungsoperation mit einem Intervall von 10 Sekunden und einem Timeout-Wert von 20 Sekunden.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.99
cidr_netmask=24 nic=eth2
```

Falls die VirtualIP einen anderen Test mit einer Tiefe von 10 unterstützt, veranlasst der folgende Befehl Packemaker dazu, alle 60 Sekunden einen umfassenderen Überwachungstest durchzuführen, zusätzlich zum normalen VirtualIP-Test alle 10 Sekunden. (Wie bereits erwähnt, sollten Sie die zusätzliche Überwachungsoperation nicht ebenfalls mit einem 10-Sekunden-Intervall konfigurieren.)

```
# pcs resource op add VirtualIP monitor interval=60s
OCF_CHECK_LEVEL=10
```

5.9. Aktivieren und Deaktivieren von Cluster-Ressourcen

Der folgende Befehl aktiviert die anhand der **resource_id** angegebene Ressource.

```
pcs resource enable resource_id
```

Der folgende Befehl deaktiviert die anhand der **resource_id** angegebene Ressource.

```
pcs resource disable resource_id
```

5.10. Bereinigen von Cluster-Ressourcen

Falls eine Ressource ausgefallen ist, erscheint eine Fehlermeldung, wenn Sie den Cluster-Status anzeigen. Falls Sie diese Ressource anschließend wiederherstellen, können Sie den Fehlerstatus mithilfe des Befehls **pcs resource cleanup** bereinigen. Dieser Befehl setzt den Status und den Ausfallzähler der Ressource zurück und weist den Cluster an, die Operationschronik einer Ressource zu vergessen und deren aktuellen Status neu zu erkennen.

Der folgende Befehl bereinigt die anhand der **resource_id** angegebene Ressource.

```
pcs resource cleanup resource_id
```

Kapitel 6. Ressourcenbeschränkungen

Sie können das Verhalten einer Ressource im Cluster bestimmen, indem Sie Ressourcenbeschränkungen für diese Ressource festlegen. Die folgenden Kategorien für Beschränkungen stehen zur Verfügung:

- **`location`** – Eine Standortbeschränkung legt fest, auf welchen Knoten eine Ressource laufen darf. Standortbeschränkungen werden in Abschnitt 6.1, „Standortbeschränkungen“ beschrieben.
- **`order`** – Eine Reihenfolgebeschränkung legt fest, in welcher Reihenfolge die Ressourcen laufen. Reihenfolgebeschränkungen werden in Abschnitt 6.2, „Reihenfolgebeschränkungen“ beschrieben.
- **`colocation`** – Eine relative Standortbeschränkung legt fest, wo eine Ressource relativ zu anderen Ressourcen platziert werden darf. Beschränkungen des relativen Standorts werden in Abschnitt 6.3, „Relative Standortbeschränkung für Ressourcen“ beschrieben.

Pacemaker unterstützt das Konzept der Ressourcengruppen als verkürzte Methode zur Konfiguration einer Reihe von Beschränkungen, die eine Gruppe von Ressourcen zusammen platziert und sicherstellt, dass die Ressourcen in der richtigen Reihenfolge starten und in der umgekehrten Reihenfolge stoppen. Weitere Informationen über Ressourcengruppen finden Sie in Abschnitt 6.5, „Ressourcenruppen“.

6.1. Standortbeschränkungen

Standortbeschränkungen legen fest, auf welchen Knoten eine Ressource laufen darf. Mithilfe von Standortbeschränkungen können Sie festlegen, ob eine Ressource einen bestimmten Knoten meidet oder bevorzugt.

Tabelle 6.1, „Optionen für Standortbeschränkungen“ zeigt eine Übersicht der Optionen zur Konfiguration von Standortbeschränkungen.

Tabelle 6.1. Optionen für Standortbeschränkungen

Feld	Beschreibung
`id`	Ein eindeutiger Name für die Beschränkung. Dieser wird vom System festgelegt, wenn Sie eine Standortbeschränkung mit **`pcs`** konfigurieren.
`rsc`	Ein Ressourcenname
`node`	Ein Knotenname
`score`	Gewichtung, die festlegt, ob eine Ressource einen Knoten bevorzugen oder meiden soll. Der Wert **`INFINITY`** ändert „soll“ auf „muss“. **`INFINITY`** ist die standardmäßige Gewichtung für eine Standortbeschränkung einer Ressource.

Der folgende Befehl erstellt eine Standortbeschränkung für eine Ressource, um den angegebenen Knoten zu bevorzugen.

```
pcs constraint location rsc prefers node[=score] ...
```

Der folgende Befehl erstellt eine Standortbeschränkung für eine Ressource, um den angegebenen Knoten zu meiden.

```
pcs constraint location rsc avoids node[=score] ...
```

Es gibt zwei alternative Strategien zur Angabe von Knoten, auf denen eine Ressource laufen soll:

- Opt-In-Cluster: Konfigurieren Sie einen Cluster, in dem standardmäßig auf keinem Knoten eine Ressource laufen darf, und aktivieren Sie anschließend einzelne erlaubte Knoten für bestimmte Ressourcen. Das Verfahren zur Konfiguration eines Opt-In-Clusters wird in Abschnitt 6.1.1, „Konfigurieren eines Opt-In-Clusters“ beschrieben.
- Opt-Out-Cluster: Konfigurieren Sie einen Cluster, in dem standardmäßig auf allen Knoten alle Ressourcen laufen dürfen, und erstellen Sie anschließend Standortbeschränkungen für Ressourcen, die nicht auf bestimmten Knoten laufen dürfen. Das Verfahren zur Konfiguration eines Opt-Out-Clusters wird in Abschnitt 6.1.2, „Konfigurieren eines Opt-Out-Clusters“ beschrieben.

Ob Sie einen Opt-In- oder Opt-Out-Cluster konfigurieren sollten, hängt sowohl von Ihren persönlichen Präferenzen als auch vom Aufbau Ihres Clusters ab. Falls die meisten Ihrer Ressourcen auf der Mehrheit Ihrer Knoten laufen können, erzielen Sie mit dem Opt-Out-Verfahren wahrscheinlich eine einfachere Konfiguration. Falls die meisten Ihrer Ressourcen jedoch nur auf einer kleinen Untergruppe von Knoten laufen können, ist eine Opt-In-Konfiguration wahrscheinlich einfacher.

6.1.1. Konfigurieren eines Opt-In-Clusters

Setzen Sie zum Erstellen eines Opt-In-Clusters die Cluster-Eigenschaft **symmetric-cluster** auf **false**, um Ressourcen daran zu hindern, irgendwo standardmäßig zu laufen.

```
# pcs property set symmetric-cluster=false
```

Aktivieren Sie Knoten für einzelne Ressourcen. Die folgenden Befehle konfigurieren Standortbeschränkungen, sodass die Ressource **Webserver** den Knoten **example-1** bevorzugt, die Ressource **Database** den Knoten **example-2** bevorzugt und beide Ressourcen auf Knoten **example-3** ausweichen können, falls ihr bevorzugter Knoten ausfällt.

```
# pcs constraint location Webserver prefers example-1=200
# pcs constraint location Webserver prefers example-3=0
# pcs constraint location Database prefers example-2=200
# pcs constraint location Database prefers example-3=0
```

6.1.2. Konfigurieren eines Opt-Out-Clusters

Setzen Sie zum Erstellen eines Opt-Out-Clusters die Cluster-Eigenschaft **symmetric-cluster** auf **true**, um Ressourcen zu erlauben, standardmäßig überall zu laufen.

```
# pcs property set symmetric-cluster=true
```

Die folgenden Befehle ergeben eine Konfiguration ähnlich des Beispiels in Abschnitt 6.1.1, „Konfigurieren eines Opt-In-Clusters“. Beide Ressourcen können auf Knoten **example-3** wechseln, falls ihr bevorzugter Knoten ausfällt, da jeder Knoten die implizite Gewichtung 0 hat.

```
# pcs constraint location Webserver prefers example-1=200
# pcs constraint location Webserver avoids example-2=INFINITY
# pcs constraint location Database avoids example-1=INFINITY
# pcs constraint location Database prefers example-2=200
```

Beachten Sie, dass es nicht notwendig ist, in diesen Befehlen die Gewichtung INFINITY anzugeben, da dies der Standardwert für die Gewichtung ist.

6.2. Reihenfolgebeschränkungen

Reihenfolgebeschränkungen bestimmen die Reihenfolge, in der die Ressourcen laufen. Sie können eine Reihenfolgebeschränkung konfigurieren, um die Reihenfolge festzulegen, in der die Ressourcen starten und stoppen.

Verwenden Sie den folgenden Befehl, um eine Reihenfolgebeschränkung zu konfigurieren.

```
pcs constraint order [action] resource_id then [action] resource_id
[options]
```

Tabelle 6.2, „Eigenschaften einer Reihenfolgebeschränkung“ zeigt eine Übersicht der Eigenschaften und Optionen zur Konfiguration von Reihenfolgebeschränkungen.

Tabelle 6.2. Eigenschaften einer Reihenfolgebeschränkung

Feld	Beschreibung
resource_id	Der Name der Ressource, auf der eine Aktion durchgeführt wird
action	Die Aktion, die auf einer Ressource durchgeführt werden soll. Mögliche Werte der *action*-Eigenschaft sind: * **start** – Startet die Ressource * **stop** – Stoppt die Ressource * **promote** – Stuft die Ressource von einer Slave-Ressource zu einer Master-Ressource hoch * **demote** – Stuft die Ressource von einer Master-Ressource zu einer Slave-Ressource zurück Falls keine Aktion angegeben wird, lautet die Standardaktion **start**. Informationen über Master- und Slave-Ressourcen finden Sie in Abschnitt 8.2, „Multi-Status-Ressourcen: Ressourcen mit mehreren Modi“
kind-Option	Wie die Beschränkung erzwungen wird. Mögliche Werte der **kind**-Option sind: * **Optional** – Nur zutreffend, falls beide Ressourcen starten bzw. stoppen. Informationen über eine optionale Reihenfolge finden Sie in Abschnitt 6.2.2, „Optionale Reihenfolge“ * **Mandatory** – Immer (Standardwert). Falls die erste angegebene Ressource gestoppt ist oder nicht starten kann, muss die zweite angegebene Ressource gestoppt werden. Informationen über die Mandatory-Reihenfolge finden Sie in Abschnitt 6.2.1, „Zwingende Reihenfolge“ * **Serialize** – Gewährleistet, dass keine zwei Stopp-/Startaktionen gleichzeitig für eine Gruppe von Ressourcen erfolgen
symmetrical-Option	Falls **true**, werden die Ressourcen in der umgekehrten Reihenfolge gestoppt. Der Standardwert ist: **true**

6.2.1. Zwingende Reihenfolge

Zwingende (**mandatory**) Beschränkungen bedeuten, dass die zweite angegebene Ressource nicht laufen kann, ohne dass die erste angegebene Ressource aktiv ist. Dies ist der Standardwert der **kind**-Option. Die Standardwerte stellen sicher, dass die zweite angegebene Ressource reagiert, wenn die erste angegebene Ressource ihren Status ändert.

- Falls die erste angegebene Ressource lief und nun gestoppt wird, wird die zweite angegebene Ressource, sofern sie läuft, ebenfalls gestoppt.
- Falls die erste angegebene Ressource nicht lief und nicht gestartet werden kann, wird die zweite angegebene Ressource, sofern sie läuft, gestoppt.
- Falls die erste angegebene Ressource (neu) gestartet wird, während die zweite angegebene Ressource läuft, wird die zweite angegebene Ressource gestoppt und neu gestartet.

6.2.2. Optionale Reihenfolge

Wenn die Option **kind=Optional** für eine Reihenfolgebeschränkung angegeben ist, dann wird die Beschränkung als optional betrachtet und greift nur dann, wenn beide Ressourcen gestoppt oder gestartet werden. Jegliche Statusänderung der ersten angegebenen Ressource hat keinerlei Auswirkung auf die zweite angegebene Ressource.

Der folgende Befehl konfiguriert eine optionale Reihenfolgebeschränkung für die Ressourcen namens **VirtualIP** und **dummy_resource**.

```
# pcs constraint VirtualIP then dummy_resource kind=Optional
```

6.2.3. Geordnete Ressourcengruppen

Ein häufiger Anwendungsfall für Administratoren ist die Erstellung einer Kette von geordneten Ressourcen, in der z. B. Ressource A vor Ressource B startet, die wiederum vor Ressource C startet. Sie können eine derartige Kette von geordneten Ressourcen mit dem folgenden Befehl erstellen. Die Ressourcen starten in der angegebenen Reihenfolge.

```
pcs constraint order set resource1 resource2 [resourceN]... [options]
[set resource1 resource2 ...]
```

Wenn Sie drei Ressourcen namens **D1**, **D2** und **D3** haben, konfiguriert der folgende Befehl sie als geordnete Ressourcengruppe.

```
# pcs constraint order set D1 D2 D3
```

6.2.4. Entfernen von Ressourcen aus Reihenfolgebeschränkungen

Verwenden Sie den folgenden Befehl, um Ressourcen aus einer Reihenfolgebeschränkung zu entfernen.

```
pcs constraint order remove resource1 [resourceN]...
```

6.3. Relative Standortbeschränkung für Ressourcen

Eine relative Standortbeschränkung legt fest, dass der Standort einer Ressource vom Standort einer anderen Ressource abhängt.

Die Erstellung von relativen Standortbeschränkungen zwischen zwei Ressourcen hat einen wichtigen Nebeneffekt: Sie hat Auswirkungen auf die Reihenfolge, in der Ressourcen einem Knoten zugewiesen werden. Dies liegt daran, dass Ressource A nicht relativ zu Ressource B platziert werden kann, bevor nicht der Standort von Ressource B bekannt ist. Bei der Konfiguration von relativen Standortbeschränkungen sollten Sie infolgedessen darauf achten, ob Sie Ressource A relativ zu Ressource B oder Ressource B relativ zu Ressource A platzieren.

Ein anderer Effekt, den es bei der Konfiguration von relativen Standortbeschränkungen zu bedenken gilt, ist der Umstand, dass der Cluster bei der Auswahl eines Knotens für Ressource B auch die Ressourcenpräferenzen von Ressource A berücksichtigt, wenn Ressource A relativ zu Ressource B platziert werden soll.

Der folgende Befehl erstellt eine relative Standortbeschränkung.

```
pcs constraint colocation add [master|slave] source_resource with
[master|slave] target_resource [score] [options]
```

Informationen über Master- und Slave-Ressourcen finden Sie in Abschnitt 8.2, „Multi-Status-Ressourcen: Ressourcen mit mehreren Modi“.

Tabelle 6.3, „Eigenschaften einer relativen Standortbeschränkung“ zeigt eine Übersicht der Eigenschaften und Optionen zur Konfiguration von relativen Standortbeschränkungen.

Tabelle 6.3. Eigenschaften einer relativen Standortbeschränkung

Feld	Beschreibung
source_resource	Die Quelle für die relative Standortbeschränkung. Falls die Bedingungen der Beschränkung nicht erfüllt werden können, kann der Cluster entscheiden, die Ressource nicht auszuführen.
target_resource	Das Ziel für die relative Standortbeschränkung. Der Cluster entscheidet zunächst über den Standort dieser Ressource, bevor er den Standort der Quellressource festlegt.
score	Positive Werte zeigen an, dass die Ressourcen auf demselben Knoten laufen sollen. Negative Werte zeigen an, dass die Ressourcen nicht auf demselben Knoten laufen sollen. Der Wert **`+INFINITY`**, der Standardwert, zeigt an, dass die *source_resource* auf demselben Knoten wie die *target_resource* laufen muss. Der Wert **`-INFINITY`** zeigt an, dass die *source_resource* nicht auf demselben Knoten wie die *target_resource* laufen darf.

6.3.1. Zwingende Platzierung

Eine zwingende Platzierung erfolgt immer dann, wenn die Gewichtung der Beschränkung **`+INFINITY`** oder **`-INFINITY`** lautet. Falls die Bedingung der Beschränkung nicht erfüllt werden kann, darf *source_resource* nicht laufen. Im Falle von **`score=INFINITY`** gehören dazu Situationen, in denen die *target_resource* nicht aktiv ist.

Angenommen, **`myresource1`** muss stets auf demselben Rechner wie **`myresource2`** laufen, dann fügen Sie die folgende Beschränkung hinzu:

```
# pcs constraint colocation add myresource1 with myresource2
score=INFINITY
```

Da **INFINITY** verwendet wurde, wird **myresource1** nicht laufen dürfen, falls **myresource2** aus irgendeinem Grund auf keinem der Cluster-Knoten laufen kann.

Alternativ können Sie das Gegenteil konfigurieren – einen Cluster, in dem **myresource1** nicht auf demselben Rechner laufen darf wie **myresource2**. Verwenden Sie in diesem Fall **score=-INFINITY**.

```
# pcs constraint colocation add myresource1 myresource2 with score=-
INFINITY
```

Durch Angabe von **-INFINITY** ist auch hier die Beschränkung zwingend. Falls der einzige verbleibende Ort zur Ausführung bereits von **myresource2** belegt ist, dann darf **myresource1** nirgends laufen.

6.3.2. Optionale Platzierung

Im Gegensatz zur zwingenden Platzierung, die ein „müssen“ und „nicht dürfen“ vorschreibt, geht es bei der optionalen Platzierung um ein „vorzugsweise“. Bei Bedingungen mit Gewichtungen größer als **-INFINITY** und kleiner als **INFINITY** versucht der Cluster, Ihre Wünsche zu erfüllen, wird diese jedoch ignorieren, falls andernfalls einige der Cluster-Ressourcen gestoppt werden müssten. Optionale, relative Standortbedingungen können mit anderen Elementen der Konfiguration kombiniert werden, um sich wie zwingende Bedingungen zu verhalten.

6.3.3. Relative Standortbeschränkung für Ressourcengruppen

Verwenden Sie den folgenden Befehl, um eine relative Standortbeschränkung auf einer Ressourcengruppe zu erstellen. Sie können die **sequential**-Option auf **true** oder **false** setzen, um anzugeben, ob es sich bei der Ressourcengruppe mit relativer Standortbeschränkung um eine geordnete Gruppe handelt.

```
colocation set resource1 resource2 [resourceN]... [setoptions name=value]
... [set resourceX resourceY ...] [setoptions name=value...]
```

Sie können die **role**-Option für eine Gruppe mit relativer Standortbeschränkung auf **master** oder **slave** setzen. Informationen über Multi-Status-Ressourcen finden Sie in Abschnitt 8.2, „Multi-Status-Ressourcen: Ressourcen mit mehreren Modi“.

6.3.4. Entfernen von relativen Standortbeschränkungen

Verwenden Sie den folgenden Befehl, um relative Standortbeschränkungen mit *source_resource* zu entfernen.

```
pcs constraint colocation remove source_resource target_resource
```

6.4. Anzeigen von Beschränkungen

Es gibt mehrere Befehle, mit denen Sie konfigurierte Beschränkungen anzeigen können.

Der folgende Befehl listet alle aktuellen Reihenfolge-, Standort- und relativen Standortbeschränkungen auf.

```
pcs constraint list|show
```

Der folgende Befehl listet alle aktuellen Standortbeschränkungen auf.

- Falls **resources** angegeben ist, werden die Standortbeschränkungen pro Ressource angezeigt. Dies ist das Standardverhalten.
- Falls **nodes** angegeben ist, werden Standortbeschränkungen pro Knoten angezeigt.
- Falls bestimmte Ressourcen oder Knoten angegeben sind, werden nur Informationen über diese Ressourcen bzw. Knoten angezeigt.

```
pcs constraint location [show resources|nodes [specific nodes|resources]]
[--full]
```

Der folgende Befehl listet alle aktuellen Reihenfolgebeschränkungen auf. Falls die Option **--full** angegeben ist, werden die internen Beschränkungs-IDs angezeigt.

```
pcs constraint order show [--full]
```

Der folgende Befehl listet alle aktuellen relativen Reihenfolgebeschränkungen auf. Falls die Option **--full** angegeben ist, werden die internen Beschränkungs-IDs angezeigt.

```
pcs constraint colocation show [--full]
```

Der folgende Befehl listet die Beschränkungen auf, die sich auf bestimmte Ressourcen beziehen.

```
pcs constraint ref resource ...
```

6.5. Ressourcenruppen

Eines der häufigsten Elemente eines Clusters ist eine Gruppe von Ressourcen, die zusammen platziert, nacheinander gestartet und in der umgekehrten Reihenfolge gestoppt werden müssen. Um diese Konfiguration zu vereinfachen, unterstützt Pacemaker das Konzept der Ressourcengruppen.

Mithilfe des folgenden Befehls erstellen Sie eine Ressourcengruppe und geben dabei die Ressourcen an, die in der Gruppe enthalten sein sollen. Falls die Gruppe noch nicht existiert, erstellt dieser Befehl die Gruppe. Falls die Gruppe bereits existiert, fügt der Befehl die zusätzlich angegebenen Ressourcen zur Gruppe hinzu. Die Ressourcen werden in der Reihenfolge starten, in der sie in diesem Befehl angegeben wurden, und werden in der umgekehrten Reihenfolge stoppen.

```
pcs resource group add group_name resource_id...
```

Mithilfe des folgenden Befehls können Sie auch eine neue Ressource während deren Erstellung zu einer vorhandenen Gruppe hinzufügen. Die erstellte Ressource wird zur Gruppe namens *group_name* hinzugefügt.

```
pcs resource create resource_id standard:provider:type|type
[resource_options] [op operation_action operation_options] --group
group_name
```

Mithilfe des folgenden Befehls entfernen Sie eine Ressource aus einer Gruppe. Falls in dieser Gruppe keine Ressourcen vorhanden sind, entfernt dieser Befehl die Gruppe selbst.

```
pcs resource group remove group_name resource_id...
```

Der folgende Befehl listet alle derzeit konfigurierten Ressourcengruppen auf.

```
pcs resource group list
```

Das folgende Beispiel erstellt eine Ressourcengruppe namens **shortcut**, welche die vorhandenen Ressourcen **IPaddr** und **Email** enthält.

```
# pcs resource group add shortcut IPaddr Email
```

Die Anzahl möglicher Ressourcen in einer Gruppe ist nicht begrenzt. Die wesentlichen Eigenschaften einer Gruppe sind wie folgt.

- Ressourcen werden in der Reihenfolge gestartet, in der Sie sie angegeben (in diesem Beispiel als Erstes **IPaddr**, danach **Email**).
- Ressourcen werden in der umgekehrten Reihenfolge gestoppt, in der Sie sie angeben (**Email** zuerst, danach **IPaddr**).

Falls eine Ressource in dieser Gruppe nirgendwo im Cluster laufen kann, dann darf auch keine der Ressourcen laufen, die nach dieser Ressource angegeben wurden.

- Falls **IPaddr** nirgends laufen darf, kann auch **Email** nirgends laufen.
- Falls **Email** nirgends laufen darf, hat dies jedoch keinerlei Auswirkungen auf **IPaddr**.

Mit wachsendem Umfang der Gruppe wird auch der eingesparte Konfigurationsaufwand bei der Erstellung von Konfigurationsgruppen deutlich.

6.5.1. Gruppenoptionen

Eine Ressourcengruppe erbt die folgenden Optionen von den darin enthaltenen Ressourcen: **priority**, **target-role**, **is-managed**. Informationen über Ressourcenoptionen finden Sie in Tabelle 5.3, „Ressourcen-Metaoptionen“.

6.5.2. Gruppentreue

Gruppentreue – ein Maß dafür, wie sehr eine Ressource bleiben möchte, wo sie ist – ist in Gruppen kumulativ. Jede aktive Ressource der Gruppe trägt seinen Treuewert zum Gesamtwert der Gruppe bei. Falls die standardmäßige Gruppentreue einer Ressource (**resource-stickiness**) 100 beträgt und eine Gruppe sieben Mitglieder hat, von denen fünf inaktiv sind, dann möchte die Gruppe insgesamt am derzeitigen Standort bleiben mit einem Treuewert von 500.

Kapitel 7. Verwalten von Cluster-Ressourcen

Dieses Kapitel beschreibt die verschiedenen Befehle, die Sie zur Verwaltung der Cluster-Ressourcen verwenden können. Es liefert Informationen zu den folgenden Verfahren.

- Abschnitt 7.1, „Manuelles Verlegen von Ressourcen im Cluster“
- Abschnitt 7.2, „Verlegen von Ressourcen wegen Ausfall“
- Abschnitt 7.4, „Aktivieren, Deaktivieren und Ausschließen von Cluster-Ressourcen“
- Abschnitt 7.5, „Deaktivieren von Überwachungsoperationen“

7.1. Manuelles Verlegen von Ressourcen im Cluster

Sie können die Einstellungen des Clusters außer Kraft setzen und Ressourcen an einen anderen Standort zwingen. Dafür gibt es zwei Anwendungsfälle:

- Wenn ein Knoten gewartet wird und Sie alle auf diesem Knoten laufenden Ressourcen auf einen anderen Knoten verlegen müssen
- Wenn eine einzelne Ressource verlegt werden muss

Um alle auf einem Knoten laufenden Ressourcen auf einen anderen Knoten zu verlegen, versetzen Sie den Knoten in den Standby-Modus. Informationen über das Versetzen eines Cluster-Knotens in den Standby-Modus finden Sie in Abschnitt 3.2.4, „Standby-Modus“.

Um eine Ressource von dem Knoten, auf dem sie derzeit läuft, wegzuverlegen, verwenden Sie den folgenden Befehl und geben Sie dabei die **resource_id** des Knotens an.

```
pcs resource move resource_id
```

Falls Sie festlegen möchten, wohin die Ressource verlegt werden soll, verwenden Sie den folgenden Befehl, um den Zielknoten mit **destination_node** anzugeben.

```
pcs resource move resource_id destination_node
```

Verwenden Sie den folgenden Befehl, um die Ressource zurück auf den Knoten zu verlegen, auf dem sie ursprünglich lief, sodass der Cluster wieder seinen normalen Betrieb aufnehmen kann. Dies entfernt die Beschränkung, die der Befehl **move resource_id** definiert hat.

```
pcs resource clear resource_id [node]
```

Beachten Sie, dass die Ausführung des Befehls **pcs resource move** der Ressource eine Beschränkung auferlegt, um diese daran zu hindern, auf dem angegebenen Knoten zu laufen. Wenn Sie den Befehl **pcs resource clear** ausführen, wird diese Beschränkung entfernt. Dadurch werden die Ressourcen nicht automatisch auf den angegebenen Knoten zurückverlegt; wo die Ressourcen zu diesem Zeitpunkt ausgeführt werden können, hängt von der ursprünglichen Konfiguration Ihrer Ressourcen ab. Informationen über Ressourcenbeschränkungen finden Sie in Kapitel 6, *Ressourcenbeschränkungen*.

7.2. Verlegen von Ressourcen wegen Ausfall

Beim Erstellen einer Ressource können Sie diese so konfigurieren, dass diese nach einer festgelegten Anzahl von Ausfällen auf einen anderen Knoten verlegt wird, indem Sie die Option **`migration-threshold`** für diese Ressource festlegen. Sobald dieser Grenzwert erreicht wurde, darf die Ressource nicht länger auf diesem Knoten ausgeführt werden, bis:

- der Administrator den Ausfallzähler der Ressource manuell mithilfe des Befehls **`pcs resource failcount`** zurücksetzt oder
- der Wert für **`failure-timeout`** für diese Ressource erreicht wird.

Standardmäßig gibt es keinen Grenzwert.

Anmerkung

Das Einstellen einer **`migration-threshold`** für eine Ressource ist nicht dasselbe wie das Konfigurieren einer Ressource für die Migration, bei der eine Ressource ohne Zustandsverlust von einem Standort auf einen anderen verlegt wird.

Das folgende Beispiel fügt einen Migrationsgrenzwert von 10 zu der Ressource namens **`dummy_resource`** hinzu, was dazu führt, dass die Ressource nach 10 Ausfällen auf einen anderen Knoten wechselt.

```
# pcs resource meta dummy_resource migration-threshold=10
```

Mithilfe des folgenden Befehls können Sie einen Migrationsgrenzwert zu den Standards für den gesamten Cluster hinzufügen.

```
# pcs resource defaults migration-threshold=10
```

Verwenden Sie den Befehl **`pcs resource failcount`**, um den aktuellen Status und die aktuellen Grenzwerte für Ausfälle für die Ressource zu bestimmen.

Es gibt zwei Ausnahmen bei Migrationsgrenzwerten: wenn der Start oder der Stopp einer Ressource fehlschlägt. Ein Fehler beim Starten führt dazu, dass der Failcount auf **`INFINITY`** gesetzt wird und die Ressource sofort verlegt wird.

Beachten Sie, dass Fehler beim Stoppen dagegen anders gehandhabt werden. Falls der Stopp einer Ressource fehlschlägt und STONITH aktiviert ist, dann grenzt der Cluster den Knoten ab, um die Ressource woanders starten zu können. Falls STONITH nicht aktiviert ist, kann der Cluster nicht fortfahren und wird nicht versuchen, die Ressource woanders zu starten. Er wird dagegen erneut versuchen, nach Ablauf der Zeitüberschreitung die Ressource zu stoppen.

7.3. Verlegen von Ressourcen wegen Änderungen der Verbindungsfähigkeit

Das Konfigurieren des Clusters zum Verlegen von Ressourcen, wenn die externe Verbindungsfähigkeit verloren ist, erfordert zwei Schritte.

1. Fügen Sie eine **`ping`**-Ressource zum Cluster hinzu. Die **`ping`**-Ressource verwendet das gleichnamige Systemdienstprogramm, um zu testen, ob eine Reihe von Rechnern (angegeben anhand deren DNS-Hostnamen oder IPv4-/IPv6-Adressen) erreichbar sind, und verwendet die Ergebnisse, um ein Knotenattribut namens **`pingd`** zu pflegen.

2. Konfigurieren Sie für die Ressource eine Standortbeschränkung, welche die Ressource auf einen anderen Knoten verlegt, falls die Verbindungsfähigkeit verloren geht.

Tabelle 5.1, „Ressourceneigenschaften“ beschreibt die Eigenschaften, die Sie für eine **ping**-Ressource festlegen können.

Tabelle 7.1. Eigenschaften von ping-Ressourcen

Feld	Beschreibung
dampen	Zeitspanne, während der auf weitere Änderungen gewartet wird. Dies verhindert, dass eine Ressource im Cluster wiederholt umherwechselt, wenn Cluster-Knoten den Verlust der Verbindungsfähigkeit leicht zeitversetzt bemerken.
multiplier	Die Anzahl der verbundenen Ping-Knoten wird mit dieser Zahl multipliziert, um eine Gewichtung zu erhalten. Dies ist hilfreich, wenn mehrere Ping-Knoten konfiguriert sind.
host_list	Den zu kontaktierenden Rechner, um den aktuellen Verbindungsstatus zu bestimmen. Zulässige Werte sind auflösbare DNS-Hostnamen und IPv4- und IPv6-Adressen.

Das folgende Beispiel erstellt eine **ping**-Ressource, welche die Verbindungsfähigkeit mit **www.example.com** prüft. In der Praxis würden Sie die Verbindungsfähigkeit zu Ihrem Netzwerk-Gateway oder Router prüfen. Konfigurieren Sie die **ping**-Ressource als Klon, damit die Ressource auf allen Cluster-Knoten läuft.

```
# pcs resource create ping ocf:pacemaker:ping dampen=5s
multiplier=1000 host_list=www.example.com --clone
```

Das folgende Beispiel konfiguriert eine Standortbeschränkungsregel für die vorhandene Ressource namens **Webserver**. Dies veranlasst die **Webserver**-Ressource dazu, auf einen Host zu wechseln, der **www.example.com** anpingen kann, falls ihr derzeitiger Host nicht mehr dazu in der Lage ist, **www.example.com** anzupingen.

```
# pcs constraint location Webserver rule score=-INFINITY pingd lt 1 or
not_defined pingd
```

7.4. Aktivieren, Deaktivieren und Ausschließen von Cluster-Ressourcen

Neben dem Befehl **pcs resource move**, der in Abschnitt 7.1, „Manuelles Verlegen von Ressourcen im Cluster“ beschrieben wird, gibt es eine Vielzahl anderer Befehle, die zur Steuerung des Verhaltens von Cluster-Ressourcen verwendet werden können.

Mithilfe des folgenden Befehls können Sie eine laufende Cluster-Ressource manuell stoppen und den Cluster daran hindern, diese wieder zu starten. Abhängig von der übrigen Konfiguration (Beschränkungen, Optionen, Ausfälle etc.) läuft die Ressource unter Umständen weiter. Falls Sie die Option **--wait** angeben, wartet **pcs** bis zu 30 Sekunden (oder „n“ Sekunden, wie festgelegt) auf den Stopp der Ressource und gibt anschließend 0 zurück, falls die Ressource gestoppt ist, oder 1, falls die Ressource nicht gestoppt wurde.

```
pcs resource disable resource_id [--wait[=n]]
```

Mithilfe des folgenden Befehls können Sie dem Cluster das Starten einer Ressource erlauben. Abhängig von der übrigen Konfiguration bleibt die Ressource unter Umständen gestoppt. Falls Sie

die Option **--wait** angeben, wartet **pcs** bis zu 30 Sekunden (oder „n“ Sekunden, wie festgelegt) auf den Start der Ressource und gibt anschließend 0 zurück, falls die Ressource gestartet ist, oder 1, falls die Ressource nicht gestartet wurde.

```
pcs resource enable resource_id [--wait[=n]]
```

Verwenden Sie den folgenden Befehl, um eine Ressource daran zu hindern, auf dem angegebenen Knoten zu laufen bzw. auf dem aktuellen Knoten zu laufen, falls kein Knoten angegeben wird.

```
pcs resource ban resource_id [node]
```

Beachten Sie, dass die Ausführung des Befehls **pcs resource ban** der Ressource eine Beschränkung auferlegt, um diese daran zu hindern, auf dem angegebenen Knoten zu laufen. Wenn Sie den Befehl **pcs resource clear** ausführen, wird diese Beschränkung entfernt. Dadurch werden die Ressourcen nicht automatisch auf den angegebenen Knoten zurückverlegt; wo die Ressourcen zu diesem Zeitpunkt ausgeführt werden können, hängt von der ursprünglichen Konfiguration Ihrer Ressourcen ab. Informationen über Ressourcenbeschränkungen finden Sie in Kapitel 6, *Ressourcenbeschränkungen*.

```
pcs resource clear resource_id [node]
```

Sie können den Parameter **debug-start** des Befehls **pcs resource** verwenden, um eine angegebene Ressource zum Starten auf dem aktuellen Knoten zu zwingen, wobei die Cluster-Empfehlungen ignoriert werden und die Ausgabe vom Ressourcenstart angezeigt wird. Dies wird hauptsächlich zur Suche und Bereinigung von Programmfehlern in Ressourcen verwendet. Das Starten von Ressourcen auf einem Cluster wird fast ausschließlich von Pacemaker gehandhabt und nicht direkt mit einem **pcs**-Befehl. Falls Ihre Ressource nicht startet, liegt dies in der Regel an einer fehlerhaften Konfiguration der Ressource (was Sie im Systemprotokoll diagnostizieren können), an Beschränkungen, die die Ressource am Starten hindern, oder daran, dass die Ressource deaktiviert ist. Mithilfe dieses Befehls können Sie die Ressourcenkonfiguration testen, er sollte jedoch normalerweise nicht zum Starten von Ressourcen in einem Cluster verwendet werden.

Das Format des Befehls **debug-start** lautet wie folgt.

```
pcs resource debug-start resource_id
```

7.5. Deaktivieren von Überwachungsoperationen

Die einfachste Methode zum Stoppen einer wiederkehrenden Überwachungsoperation ist das Löschen derselben. Allerdings ist es in einigen Situationen ggf. sinnvoller, diese nur vorübergehend zu deaktivieren. Fügen Sie dazu **enabled="false"** zur Definition der Operation hinzu. Wenn Sie die Überwachungsoperation wieder aktivieren möchten, fügen Sie **enabled="true"** zur Definition der Operation hinzu.

7.6. Verwaltete Ressourcen

Sie können eine Ressource in einen nicht verwalteten Modus versetzen. Dadurch verbleibt die Ressource in der Konfiguration, wird allerdings nicht von Pacemaker verwaltet.

Der folgende Befehl versetzt die angegebenen Ressourcen in den nicht verwalteten Modus.

```
pcs resource unmanage resource1  [resource2] ...
```

Der folgende Befehl versetzt Ressourcen in den verwalteten Modus, was der standardmäßige Status ist.

```
pcs resource manage resource1  [resource2] ...
```

Sie können zu dem Befehl **pcs resource manage** oder **pcs resource unmanage** den Namen einer Ressourcengruppe angeben. Der Befehl wirkt sich auf alle Ressourcen in der Gruppe aus, sodass Sie alle Ressourcen in der Gruppe mit einem einzigen Befehl in einen verwalteten oder nicht verwalteten Modus versetzen können, um dann die enthaltenen Ressourcen einzeln zu verwalten.

Kapitel 8. Erweiterte Ressourcentypen

Dieses Kapitel beschreibt erweiterte Ressourcentypen, die von Pacemaker unterstützt werden.

8.1. Ressourcen-Klone

Sie können eine Ressource klonen, sodass die Ressource auf mehreren Knoten aktiv sein kann. Beispielsweise können Sie geklonte Ressourcen verwenden, um mehrere Instanzen einer IP-Ressource zu konfigurieren und diese zwecks Lastverteilung im Cluster zu verteilen. Sie können jede beliebige Ressource klonen, vorausgesetzt, der Ressourcenagent unterstützt dies. Ein Klon besteht aus einer Ressource oder einer Ressourcengruppe.

Anmerkung

Nur Ressourcen, die auf mehreren Knoten gleichzeitig aktiv sein können, sind für das Klonen geeignet. Beispielsweise sollte eine **`Filesystem`**-Ressource, die ein nicht geclustertes Dateisystem wie **`ext4`** auf einem gemeinsam verwendeten Speichergerät einhängt, nicht geklont werden. Da die **`ext4`**-Partition nicht clusterfähig ist, ist das Dateisystem nicht geeignet für Lese- und Schreiboperationen, die von mehreren Knoten gleichzeitig erfolgen.

8.1.1. Erstellen und Entfernen einer geklonten Ressource

Mithilfe des folgenden Befehls können Sie gleichzeitig eine Ressource und einen Klon dieser Ressource erstellen.

```
pcs resource create resource_id standard:provider:type|type [resource
options]  \
--clone  [meta clone_options]
```

Der Name des Klons lautet ***`resource_id`*****`-clone`**.

Sie können jedoch nicht gleichzeitig eine Ressourcengruppe und einen Klon dieser Ressourcengruppe mit einem einzigen Befehl erstellen.

Alternativ können Sie mithilfe des folgenden Befehls einen Klon einer zuvor erstellen Ressource oder Ressourcengruppe erstellen.

```
pcs resource clone resource_id | group_name [clone_options]...
```

Der Name des Klons lautet ***`resource_id`*****`-clone`** oder ***`group_name`*****`-clone`**.



Anmerkung

Sie brauchen Änderungen an der Ressourcenkonfiguration nur an einem Knoten vorzunehmen.



Anmerkung

Verwenden Sie bei der Konfiguration von Beschränkungen stets den Namen der Gruppe oder des Klons.

Wenn Sie einen Klon einer Ressource erstellen, trägt dieser Klon den Namen der Ressource mit nachgestelltem **`-clone`**. Der folgende Befehl erstellt eine Ressource vom Typ **`apache`** namens **`webfarm`** und ein Klon dieser Ressource namens **`webfarm-clone`**.

```
# pcs resource create webfarm apache clone
```

Verwenden Sie den folgenden Befehl, um den Klon einer Ressource oder Ressourcengruppe zu entfernen. Die Ressource oder die Ressourcengruppe selbst wird dadurch nicht entfernt.

```
pcs resource unclone resource_id | group_name
```

Informationen über Ressourcenoptionen finden Sie in Abschnitt 5.1, „Ressourcenerstellung“.

Tabelle 8.1, „Optionen für Ressourenklone“ beschreibt die Optionen, die Sie für eine geklonte Ressource festlegen können.

Tabelle 8.1. Optionen für Ressourenklone

Feld	Beschreibung
`priority, target-role, is-managed`	Optionen, die von der zu klonenden Ressource geerbt werden, wie in Tabelle 5.3, „Ressourcen-Metaoptionen“ beschrieben.
`clone-max`	Legt die Anzahl der zu startenden Instanzen der Ressource fest. Entspricht standardmäßig der Anzahl der Knoten im Cluster.
`clone-node-max`	Legt die Anzahl der Ressourcen fest, die auf einem einzigen Knoten gestartet werden können. Der Standarwert ist **`1`**.
`notify`	Legt fest, ob beim Stoppen und Starten eines Klons alle anderen Klone vor der Aktion sowie nach erfolgter Aktion benachrichtigt werden sollen. Zulässige Werte: **`false`**, **`true`**. Der Standardwert ist **`false`**.
`globally-unique`	Legt fest, ob jeder Klon eine andere Funktion ausübt. Zulässige Werte **`false`**, **`true`** Falls der Wert dieser Option **`false`** ist, verhalten sich diese Ressourcen überall gleich, ungeachtet dessen, wo sie ausgeführt werden. Demnach kann in diesem Fall pro Rechner nur eine Instanz des Klons aktiv sein. Falls der Wert dieser Option **`true`** ist, dann ist ein Klon auf einem Rechner nicht identisch mit einer anderen Instanz des Klons auf einem anderen Knoten oder auf demselben Knoten. Der Standardwert lautet **`true`**, falls der Wert von **`clone-node-max`** größer als 1 ist, andernfalls lautet der Standardwert **`false`**.
`ordered`	Legt fest, ob die Klone nacheinander (statt gleichzeitig) gestartet werden sollen. Zulässige Werte: **`false`**, **`true`**. Der Standardwert ist **`false`**.
`interleave`	Ändert das Verhalten hinsichtlich der Reihenfolge (zwischen Klonen und Master), sodass die Instanzen starten bzw. stoppen können, sobald ihre Peer-Instanz gestartet bzw. gestoppt wurde (statt auf jede Instanz des Klons zu warten). Zulässige Werte: **`false`**, **`true`**. Der Standardwert ist **`false`**.

8.1.2. Klonbeschränkungen

In den meisten Fällen hat ein Klon eine Instanz auf jedem aktiven Cluster-Knoten. Sie können jedoch **`clone-max`** für den Ressourcenklon auf einen Wert festlegen, der geringer ist als die Gesamtanzahl der Knoten im Cluster. In diesem Fall können Sie mithilfe von Standortbeschränkungen für die Ressource angeben, welchen Knoten der Cluster diese Klone vorzugsweise zuweisen soll. Diese Beschränkungen werden analog zur jenen für reguläre Ressourcen angegeben, jedoch mit dem Unterschied, dass die Klon-ID verwendet werden muss.

Der folgende Befehl erstellt eine Standortbeschränkung, damit der Cluster den Ressourcenklon **`webfarm-clone`** vorzugsweise **`node1`** zuweist.

```
# pcs constraint location webfarm-clone prefers node1
```

Reihenfolgebeschränkungen verhalten sich etwas anders für Klone. In dem nachfolgenden Beispiel wartet **`webfarm-stats`**, bis alle Instanzen von **`webfarm-clone`**, die gestartet werden müssen, gestartet wurden, bevor er selbst startet. Nur wenn keine Instanzen von **`webfarm-clone`** gestartet werden können, hindert dies **`webfarm-stats`** daran, selbst gestartet zu werden. Umgekehrt wartet **`webfarm-clone`**, bis **`webfarm-stats`** gestoppt wurde, bevor dieser selbst stoppt.

```
# pcs constraint order start webfarm-clone then webfarm-stats
```

Eine relative Standortbeschränkung einer regulären oder Gruppenressource mit einem Klon bedeutet, dass die Ressource auf jedem Rechner mit einer aktiven Instanz des Klons laufen kann. Der Cluster wählt eine Instanz basierend darauf, wo der Klon läuft, und auf den Standortpräferenzen der Ressource selbst.

Es ist ebenfalls möglich, relative Standortbeschränkungen zwischen Klons einzurichten. In solchen Fällen wird die Liste der erlaubten Standorte für den Klon auf Knoten beschränkt, auf denen der Klon aktiv ist bzw. aktiv sein wird. Die Zuweisung erfolgt dann ganz normal.

Der folgende Befehl erstellt eine relative Standortbeschränkung, um sicherzustellen, dass die Ressource **`webfarm-stats`** auf demselben Knoten läuft wie eine aktive Instanz von **`webfarm-clone`**.

```
# pcs constraint colocation add webfarm-stats with webfarm-clone
```

8.1.3. Standorttreue von Klons

Um ein stabiles Zuweisungsmuster zu erreichen, sind Klone standardmäßig leicht standorttreu. Falls für **`resource-stickiness`** kein Wert angegeben wird, verwendet der Klon den Wert 1. Da dies nur ein kleiner Wert ist, hat er geringe Störwirkung bei der Kalkulation der Gewichtung anderer Ressourcen; dieser Wert ist jedoch genug, um Pacemaker daran zu hindern, unnötig Klone im Cluster hin- und herzuverlegen.

8.2. Multi-Status-Ressourcen: Ressourcen mit mehreren Modi

Multi-Status-Ressourcen sind spezielle Klonressourcen. Sie erlauben den Instanzen die Ausführung in einem von zwei Betriebsmodi, entweder **`Master`** oder **`Slave`**. Die Namen der Modi haben keine weitere Bewandtnis; es ist lediglich vorgeschrieben, dass eine Instanz im **`Slave`**-Status starten muss.

Mit dem folgenden Befehl können Sie eine Ressource als Master-/Slave-Klon erstellen.

```
pcs resource create resource_id standard:provider:type|type [resource
options] \
--master [meta master_options]
```

Der Name des Master-/Slave-Klons lautet ***resource_id*-master**.

Alternativ können Sie mithilfe des folgenden Befehls eine Master-/Slave-Ressource einer zuvor erstellten Ressource oder Ressourcengruppe erstellen. Wenn Sie diesen Befehl verwenden, können Sie einen Namen für den Master-/Slave-Klon angeben. Falls Sie keinen Namen angeben, lautet der Name des Master-/Slave-Klons ***resource_id*-master** bzw. ***group_name*-master**.

```
pcs resource master master/slave_name resource_id|group_name
[master_options]
```

Informationen über Ressourcenoptionen finden Sie in Abschnitt 5.1, „Ressourcenerstellung“.

Tabelle 8.2, „Eigenschaften einer Multi-Status-Ressource“ beschreibt die Optionen, die Sie für eine Multi-Status-Ressource festlegen können.

Tabelle 8.2. Eigenschaften einer Multi-Status-Ressource

Feld	Beschreibung
id	Der Name Ihrer Multi-Status-Ressource
priority, **target-role**, **is-managed**	Siehe Tabelle 5.3, „Ressourcen-Metaoptionen“.
clone-max, **clone-node-max**, **notify**, **globally-unique**, **ordered**, **interleave**	Siehe Tabelle 8.1, „Optionen für Ressourenklone“.
master-max	Wie viele Instanzen der Ressource können in den **master**-Status hochgestuft werden; standardmäßig 1.
master-node-max	Wie viele Instanzen der Ressource können auf einem einzigen Knoten in den **master**-Status hochgestuft werden; standardmäßig 1.

8.2.1. Überwachen von Multi-Status-Ressourcen

Um eine Überwachungsoperation nur für die Master-Ressource hinzuzufügen, können Sie eine zusätzliche Überwachungsoperation zur Ressource hinzufügen. Beachten Sie jedoch, dass jede Überwachungsoperation auf einer Ressource ein anderes Intervall haben muss.

Das folgende Beispiel konfiguriert eine Überwachungsoperation mit einem Intervall von 11 Sekunden auf der Master-Ressource für **ms_resource**. Diese Überwachungsoperation läuft zusätzlich zur standardmäßigen Überwachungsoperation mit dem standardmäßigen Überwachungsintervall von 10 Sekunden.

```
# pcs resource op add ms_resource interval=11s role=Master
```

8.2.2. Multi-Status-Beschränkungen

In den meisten Fällen hat eine Multi-Status-Ressource eine einzelne Instanz auf jedem aktiven Cluster-Knoten. Ist dies nicht der Fall, können Sie mithilfe von Standortbeschränkungen der Ressource angeben, welchen Knoten der Cluster diese Klone vorzugsweise zuweisen soll. Diese Beschränkungen werden analog zur jenen für reguläre Ressourcen geschrieben.

Weitere Informationen über Standortbeschränkungen für Ressourcen finden Sie in Abschnitt 6.1, „Standortbeschränkungen“.

Sie können eine relative Standortbeschränkung erstellen, die festlegt, ob es sich bei den Ressourcen um Master- oder Slave-Ressourcen handelt. Der folgende Befehl erstellt eine relative Standortbeschränkung für Ressourcen.

```
pcs constraint colocation add [master|slave] source_resource with
[master|slave] target_resource [score] [options]
```

Weitere Informationen über relative Standortbeschränkungen für Ressourcen finden Sie in Abschnitt 6.3, „Relative Standortbeschränkung für Ressourcen“.

Bei der Konfiguration einer Reihenfolgebeschränkung, die Multi-Status-Ressourcen umfasst, lautet eine der Aktionen, die Sie für die Ressourcen angeben können, **promote** – dies gibt an, dass die Ressource vom Slave zum Master hochgestuft werden soll. Sie können auch die Aktion **demote** angeben – dies gibt an, dass die Ressource vom Master zum Slave heruntergestuft werden soll.

Der Befehl zur Konfiguration einer Reihenfolgebeschränkung lautet wie folgt.

```
pcs constraint order [action] resource_id then [action] resource_id
[options]
```

Weitere Informationen über Reihenfolgebeschränkungen für Ressourcen finden Sie in Abschnitt 6.2, „Reihenfolgebeschränkungen“.

8.2.3. Multi-Status-Standorttreue

Um ein stabiles Zuweisungsmuster zu erreichen, sind Multi-Status-Ressourcen standardmäßig eher standorttreu. Falls für **resource-stickiness** kein Wert angegeben wird, verwendet die Multi-Status-Ressource den Wert 1. Da dies nur ein kleiner Wert ist, hat er geringe Störwirkung bei der Kalkulation der Gewichtung anderer Ressourcen; dieser Wert ist jedoch genug, um Pacemaker daran zu hindern, unnötig Klone im Cluster hin- und herzuverlegen.

8.3. Ereignisbenachrichtigungen mit Überwachungsressourcen

Ein Pacemaker-Cluster ist ein ereignisgesteuertes System, wobei ein Ereignis beispielsweise ein Ausfall einer Ressource oder eine Änderung in der Konfiguration sein kann. Die Ressource **ocf:pacemaker:ClusterMon** kann den Status des Clusters überwachen und bei jedem Cluster-Ereignis Benachrichtigungen ausgeben. Diese Ressource führt in regelmäßigen Abständen **crm_mon** im Hintergrund aus und verwendet Funktionen von **crm_mon** zum Senden von E-Mail-Nachrichten (SMTP) oder SNMP-Traps. Sie kann mithilfe des Parameters **extra_options** auch externe Programme ausführen.

Das folgende Beispiel konfiguriert eine **ClusterMon**-Ressource namens **ClusterMon-SMTP**, die E-Mail-Benachrichtigungen sendet. Pacemaker-Ereignisse veranlassen den Versand einer E-Mail an **pacemaker@example.com** von **pacemaker@nodeX.example.com** mit **mail.example.com** als Mail-Host. Diese Ressource wird als Klon erstellt, damit sie auf jedem Knoten im Cluster läuft.

```
# pcs resource create ClusterMon-SMTP ClusterMon --clone user=root
update=30 \ extra_options="-T pacemaker@example.com -F
pacemaker@nodeX.example.com \ -P PACEMAKER -H mail.example.com"
```

Das folgende Beispiel konfiguriert eine **ClusterMon**-Ressource namens **ClusterMon-SNMP**, die SNMP-Traps mit dem root-SNMP-Benutzer an Host **snmphost.example.com** sendet. Diese Ressource wird als Klon erstellt, damit sie auf jedem Knoten im Cluster läuft.

```
# pcs resource create ClusterMon-SNMP ClusterMon user=root update=30 \
extra_options="-S snmphost.example.com -C public" --clone
```

Das folgende Beispiel konfiguriert eine **ClusterMon**-Ressource namens **ClusterMon-External**, die das Programm **/usr/local/bin/example.sh** ausführt, das bestimmt, wie mit Cluster-Benachrichtigungen umzugehen ist. Diese Ressource wird als Klon erstellt, damit sie auf jedem Knoten im Cluster läuft.

```
# pcs resource create ClusterMon-External ClusterMon --clone user=root \
update=30 extra_options="-E /usr/local/bin/example.sh -e 192.168.12.1"
```

8.4. Der Dienst pacemaker_remote

Der Dienst **pacemaker_remote** ermöglicht es, Knoten, die **corosync** nicht ausführen, in den Cluster zu integrieren und deren Ressourcen zu verwalten, als handele es sich um echte Cluster-Knoten. Dies bedeutet, dass Pacemaker-Cluster nun dazu in der Lage sind, virtuelle Umgebungen (KVM/LXC) sowie die Ressourcen innerhalb dieser Umgebungen zu verwalten, ohne dass in diesen virtuellen Umgebungen **pacemaker** oder **corosync** ausgeführt werden muss.

Die folgenden Begriffe beschreiben den Dienst **pacemaker_remote**.

- *Cluster-Knoten* – Ein Knoten, der die Hochverfügbarkeitsdienste (**pacemaker** und **corosync**) ausführt.
- *Remote-Knoten* – Ein Knoten, der **pacemaker_remote** ausführt, um sich von Remote aus in den Cluster zu integrieren, ohne eine **corosync**-Cluster-Mitgliedschaft zu erfordern.
- *Container* – Eine Pacemaker-Ressource, die weitere Ressourcen enthält. Beispielsweise eine virtuelle KVM-Maschine, die eine Webserver-Ressource enthält.
- *Container-Remote-Knoten* – Ein virtueller Remote-Gast-Knoten, der den Dienst **pacemaker_remote** ausführt. Dies beschreibt einen bestimmten Anwendungsfall eines Remote-Knotens, bei dem eine vom Cluster verwaltete virtuelle Gastressource vom Cluster gestartet und in den Cluster als Remote-Knoten integriert wird.
- *pacemaker_remote* – Ein Dienst-Daemon mit der Fähigkeit zur Remote-Applikationsverwaltung innerhalb von Gastknoten (KVM und LXC) in sowohl Pacemaker-Cluster-Umgebungen als auch eigenständigen (nicht geclusterten) Umgebungen. Dieser Dienst ist eine erweiterte Version von Pacemakers lokalem Daemon zur Ressourcenverwaltung (Local Resource Management Daemon oder kurz LRMD), der fähig ist zur Remote-Verwaltung und -Überwachung von LSB-, OCF-, upstart- und systemd-Ressourcen auf einem Gast. Zudem ermöglicht er, dass **pcs** auf Remote-Knoten ausgeführt werden kann.
- *LXC* – Ein Linux-Container, definiert durch den Linux-Container-Treiber **libvirt-lxc**.

Ein Pacemaker-Cluster, der den Dienst **pacemaker_remote** ausführt, hat die folgenden Charakteristiken.

- Die virtuellen Remote-Knoten führen den Dienst **pacemaker_remote** aus (mit nur wenig erforderlicher Konfiguration auf Seiten der virtuellen Maschine).

- Der Cluster-Stack (**pacemaker** und **corosync**), der auf den Cluster-Knoten läuft, startet die virtuellen Maschinen und verbindet sofort mit dem Dienst **pacemaker_remote**, sodass die virtuellen Maschinen in den Cluster integriert werden können.

Der wesentliche Unterschied zwischen den virtuellen Remote-Maschinen und den Cluster-Knoten besteht darin, dass die Remote-Knoten nicht den Cluster-Stack ausführen. Dies bedeutet, dass die Remote-Knoten keinen Anteil am Quorum haben. Andererseits bedeutet dies auch, dass die Remote-Knoten nicht der begrenzten Skalierbarkeit des Cluster-Stacks unterliegen. Abgesehen von der Quorum-Einschränkung verhalten sich Remote-Knoten im Hinblick auf die Ressourcenverwaltung ganz wie Cluster-Knoten. Der Cluster ist dazu in der Lage, die Ressourcen auf jedem Remote-Knoten vollständig zu verwalten und zu überwachen. Sie können Beschränkungen für Remote-Knoten anlegen, sie in den Standby-Modus versetzen oder jede andere Aktion auf ihnen ausführen, die Sie auch auf Cluster-Knoten ausführen können. Remote-Knoten erscheinen in der Ausgabe des Cluster-Status ganz wie normale Cluster-Knoten.

8.4.1. Ressourcenoptionen für Container-Remote-Knoten

Zur Konfiguration einer virtuellen Maschine oder LXC-Ressource als Remote-Knoten erstellen Sie eine **VirtualDomain**-Ressource, welche die virtuelle Maschine verwaltet. Mit dem folgenden Befehl erhalten Sie eine Beschreibung der Optionen, die Sie für eine **VirtualDomain**-Ressource festlegen können.

```
# pcs resource describe VirtualDomain
```

Zusätzlich zu den **VirtualDomain**-Ressourcenoptionen können Sie Metadaten-Optionen konfigurieren, um die Ressource als Remote-Knoten zu konfigurieren und die Verbindungsparameter zu definieren. Tabelle 8.3, „Metadaten-Optionen zur Konfiguration von KVM/LXC-Rsesourcen als Remote-Knoten“ beschreibt diese Metadaten-Optionen.

Tabelle 8.3. Metadaten-Optionen zur Konfiguration von KVM/LXC-Rsesourcen als Remote-Knoten

Feld	Standard	Beschreibung
remote-node	<none>	Der Name des Remote-Knotens, der diese Ressource definiert. Dies aktiviert die Ressource als Remote-Knoten und definiert den eindeutigen Namen zur Identifizierung des Remote-Knotens. Falls keine Parameter festgelegt sind, wird dieser Wert auch als Hostname zur Verbindung mit Port 3121 angenommen. *WARNUNG*: Dieser Wert darf sich nicht mit anderen Ressourcen- oder Knoten-IDs überschneiden.
remote-port	3121	Konfiguriert einen benutzerdefinierten Port zur Verwendung für die Gastverbindung mit **pacemaker_remote**.
remote-addr	Wert von **remote-node** verwendet als Hostname	Die IP-Adresse oder der Hostname, mit dem verbunden wird, falls der Name des Remote-Knotens nicht der Hostname des Gasts ist
remote-connect-timeout	60s	Zeitspanne für eine ausstehende Gastverbindung, bevor eine Zeitüberschreitung erfolgt

Der folgende Befehl erstellt eine **VirtualDomain**-Ressource namens **vm-guest1**, bei der es sich um einen Remote-Knoten handelt, der mithilfe des Meta-Attributs **remote-node** dazu in der Lage ist, Ressourcen auszuführen.

```
# pcs resource create vm-guest1 VirtualDomain
hypervisor="qemu:///system" config="vm-guest1.xml" meta remote-
node=guest1
```

8.4.2. Host- und Gastauthentifizierung

Die Authentifizierung und Verschlüsselung der Verbindung zwischen Cluster-Knoten und Remote-Knoten erfolgt mittels TLS mit PSK-Verschlüsselung/Authentifizierung auf TCP-Port 3121. Das bedeutet, dass sowohl der Cluster-Knoten als auch der Remote-Knoten über denselben privaten Schlüssel verfügen müssen. Standardmäßig muss dieser Schlüssel auf den Cluster-Knoten und den Remote-Knoten unter **/etc/pacemaker/authkey** abgelegt werden.

8.4.3. Ändern der standardmäßigen pacemaker_remote-Optionen

Falls Sie den standardmäßigen Port oder Speicherort von **authkey** für entweder Pacemaker oder **pacemaker_remote** ändern möchten, gibt es Umgebungsvariablen zu diesem Zweck, die diese beiden Daemons betreffen. Sie können diese Umgebungsvariablen aktivieren, indem Sie sie wie folgt in die Datei **/etc/sysconfig/pacemaker** einfügen.

```
#==#==# Pacemaker Remote
# Use a custom directory for finding the authkey.
PCMK_authkey_location=/etc/pacemaker/authkey
#
# Specify a custom port for Pacemaker Remote connections
PCMK_remote_port=3121
```

8.4.4. Konfigurationsübersicht: KVM-Remote-Knoten

Dieser Abschnitt liefert eine Übersicht über die notwendigen Schritte, um mit Pacemaker eine virtuelle Maschine zu starten und diese als Remote-Knoten zu integrieren, unter Verwendung von **libvirt** und virtuellen KVM-Gästen.

1. Nachdem Sie die Virtualisierungssoftware installiert und den **libvirtd**-Dienst auf den Cluster-Knoten aktiviert haben, platzieren Sie einen **authkey** mit dem Pfad **/etc/pacemaker/authkey** auf jedem Cluster-Knoten und auf jeder virtuellen Maschine. Dies sichert die Remote-Kommunikation und die Authentifizierung.

   Der folgende Befehl erstellt einen **authkey**.

   ```
   # dd if=/dev/urandom of=/etc/pacemaker/authkey bs=4096 count=1
   ```

2. Installieren Sie auf jeder virtuellen Maschine die **pacemaker_remote**-Pakete, starten Sie den **pacemaker_remote**-Dienst, aktivieren Sie dessen Start beim Systemboot und öffnen Sie den TCP-Port 3121 in der Firewall.

   ```
   # yum install pacemaker-remote resource-agents
   # systemctl start pacemaker_remote.service
   # systemctl enable pacemaker_remote.service
   # firewall-cmd --add-port 3121/tcp --permanent
   ```

3. Geben Sie jeder virtuellen Maschine eine statische Netzwerkadresse und einen eindeutigen Hostnamen.

4. Um den **VirtualDomain**-Ressourcen-Agent zur Verwaltung der virtuellen Maschine zu erstellen, erfordert Pacemaker, dass ein Auszug der XML-Konfigurationsdatei der virtuellen Maschine in einer Datei auf der Festplatte gespeichert wird. Falls Sie beispielsweise eine virtuelle Maschine namens **guest1** erstellt haben, speichern Sie mithilfe des folgenden Befehls das XML in einer Datei auf dem Host ab.

```
# virsh dumpxml guest1 > /virtual_machines/guest1.xml
```

5. Erstellen Sie die **VirtualDomain**-Ressource, wobei die Ressourcen-Metaoption **remote-node** angeben sollte, dass die virtuelle Maschine ein Remote-Knoten fähig zur Ausführung von Ressourcen ist.

   In dem Beispiel unten teilt das Meta-Attribut **remote-node=guest1** Pacemaker mit, dass diese Ressource ein Remote-Knoten mit dem Hostnamen **guest1** ist, der in den Cluster integriert werden kann. Der Cluster wird versuchen, den Dienst **pacemaker_remote** der virtuellen Maschine unter dem Hostnamen **guest1** zu kontaktieren, nachdem er gestartet wurde.

```
# pcs resource create vm-guest1 VirtualDomain
hypervisor="qemu:///system" config="vm-guest1.xml" meta remote-
node=guest1
```

6. Nach Erstellung der **VirtualDomain**-Ressource können Sie diesen Remote-Knoten wie jeden anderen Knoten im Cluster behandeln. Beispielsweise können Sie eine Ressource erstellen und diese Ressource mit einer Ressourcenbeschränkung belegen, damit diese nur auf dem Remote-Knoten läuft.

```
# pcs resource create webserver apache params
configfile=/etc/httpd/conf/httpd.conf op monitor interval=30s
# pcs constraint webserver prefers guest1
```

   Sobald ein Remote-Knoten in den Cluster integriert wurde, können Sie **pcs**-Befehle auf dem Remote-Knoten selbst ausführen, ganz so, als liefe Pacemaker auf dem Remote-Knoten.

Kapitel 9. Pacemaker-Regeln

Mithilfe von Regeln können Sie Ihre Konfiguration dynamischer gestalten. Ein häufiges Anwendungsbeispiel ist das Festlegen eines Werts für **`resource-stickiness`** während der Geschäftszeiten, um Ressourcen daran zu hindern, auf Ihre bevorzugten Standorte zurückzuwechseln, und einen anderen Wert für Wochenenden, wenn niemand etwaige Serviceausfälle bemerkt.

Ein anderer Anwendungsfall für Regeln ist das Zuweisen von Rechnern zu verschiedenen Rechengruppen (mithilfe eines Knotenattributs) basierend auf Zeit, um dann dieses Attribut bei der Erstellung von Standortbeschränkungen zu verwenden.

Jede Regel kann eine Reihe von Ausdrücken, Datumsausdrücken sowie andere Regeln enthalten. Die Ergebnisse der Ausdrücke werden kombiniert basierend auf dem booleschen Operator **`boolean-op`**, um zu bestimmen, ob die Regel letztlich **`true`** oder **`false`** ergibt. Was danach geschieht, hängt vom Zusammenhang ab, in dem die Regel verwendet wird.

Tabelle 9.1. Eigenschaften einer Regel

Feld	Beschreibung
`role`	Legt fest, dass die Regel nur dann angewendet wird, falls die Ressource diese Rolle einnimmt. Zulässige Werte: **`Started`**, **`Slave`** und **`Master`**. HINWEIS: Eine Regel mit **`role="Master"`** kann nicht den ursprünglichen Standort einer Kloninstanz bestimmen. Sie wirkt sich nur darauf aus, welche der aktiven Instanzen hochgestuft werden.
`score`	Die anzuwendende Gewichtung, wenn die Regel **`true`** ergibt. Kann nur in Regeln verwendet werden, die Teil von Standortbeschränkungen sind.
`score-attribute`	Das abzurufende und als Gewichtung zu verwendende Knotenattribut, falls die Regel **`true`** ergibt. Kann nur in Regeln verwendet werden, die Teil von Standortbeschränkungen sind.
`boolean-op`	Wie die Ergebnisse mehrerer Ausdrucksobjekte kombiniert werden sollen. Zulässige Werte: **`and`** und **`or`**. Der Standardwert ist **`and`**.

9.1. Knotenattribut-Ausdrücke

Knotenattribut-Ausdrücke werden verwendet, um eine Ressource basierend auf den Attributen zu steuern, die von den Knoten definiert sind.

Tabelle 9.2. Eigenschaften eines Ausdrucks

Feld	Beschreibung
`value`	Ein benutzerdefinierter Wert zum Vergleich
`attribute`	Ein Knotenattribut zum Testen
`type`	Bestimmt, wie die Werte getestet werden sollen. Zulässige Werte: **`string`**, **`integer`**, **`version`**

Feld	Beschreibung
operation	Der auszuführende Vergleich. Zulässige Werte: * **lt** – Ergibt **true**, falls der Wert des Knotenattributs weniger als **value** beträgt * **gt** – Ergibt **true**, falls der Wert des Knotenattributs größer als **value** ist * **lte** – Ergibt **true**, falls der Wert des Knotenattributs weniger als oder gleich **value** ist * **gte** – Ergibt **true**, falls der Wert des Knotenattributs größer als oder gleich **value** ist * **eq** – Ergibt **true**, falls der Wert des Knotenattributs gleich **value** ist * **ne** – Ergibt **true**, falls der Wert des Knotenattributs nicht gleich **value** ist * **defined** – Ergibt **true**, falls der Knoten das genannte Attribut hat * **defined** – Ergibt **true**, falls der Knoten nicht das genannte Attribut hat

9.2. Zeit-/Datumsbasierte Ausdrücke

Datumsausdrücke werden verwendet, um eine Ressource oder Cluster-Option basierend auf dem aktuellen Datum oder der aktuellen Zeit zu steuern. Sie können eine optionale Datumsspezifikation enthalten.

Tabelle 9.3. Eigenschaften eines Datumsausdrucks

Feld	Beschreibung
start	Eine Datums-/Zeitangabe nach ISO8601-Spezifikation
end	Eine Datums-/Zeitangabe nach ISO8601-Spezifikation
operation	Vergleicht das aktuelle Datum und die aktuelle Uhrzeit mit dem Start- und/oder dem Enddatum, abhängig vom Kontext. Zulässige Werte: * **gt** – Ergibt **true**, falls das aktuelle Datum und die aktuelle Uhrzeit nach **start** liegt * **lt** – Ergibt **true**, falls das aktuelle Datum und die aktuelle Uhrzeit vor **end** liegt * **in-range** – Ergibt **true**, falls das aktuelle Datum und die aktuelle Uhrzeit nach **start** und vor **end** liegt * **date-spec** – Führt einen cron-artigen Vergleich zum aktuellen Datum und zur aktuellen Zeit durch

9.3. Datumsspezifikationen

Datumsspezifikationen werden dazu verwendet, um cron-artige Ausdrücke der Zeit zu erstellen. Jedes Feld kann eine einzelne Zahl oder einen einzelnen Bereich enthalten. Wird ein Feld nicht spezifiziert, wird dafür nicht der Standardwert Null angenommen, sondern das Feld ignoriert.

Beispielsweise bedeutet **monthdays="1"** der erste Tag eines jeden Monats und **hours="09-17"** bedeutet die Stunden zwischen 9 Uhr morgens und 17 Uhr abends (einschließlich). Allerdings können Sie nicht **weekdays="1,2"** oder **weekdays="1-2,5-6"** angeben, da dies mehrere Bereiche darstellt.

Tabelle 9.4. Eigenschaften einer Datumsspezifikation

Feld	Beschreibung
id	Ein eindeutiger Name für das Datum
hours	Zulässige Werte: 0–23
monthdays	Zulässige Werte: 0–31 (abhängig vom Monat und Jahr)
weekdays	Zulässige Werte: 1–7 (1=Montag, 7=Sonntag)
yeardays	Zulässige Werte: 0–366 (abhängig vom Jahr)
months	Zulässige Werte: 0–12
weeks	Zulässige Werte: 0–53 (abhängig von weekyear)
years	Jahr nach Gregorianischem Kalender
weekyears	Gegebenenfalls abweichend von Gregorianischem Jahren, zum Beispiel entspricht **2005-001 Ordinal** der Angabe **2005-01-01 Gregorian** und der Angabe **2004-W53-6 Weekly**
moon	Zulässige Werte: 0–7 (0=Neumond, 4=Vollmond)

9.4. Dauer

Die Dauer wird zur Berechnung eines Werts für **end** verwendet, wenn in **in_range**-Operationen keiner angegeben wurde. Die Dauer verwendet dieselben Felder wie **date_spec**-Objekte, jedoch ohne Begrenzungen (z. B. können Sie eine Dauer von 19 Monaten festlegen). Wie bei **date_specs** wird auch hier jedes nicht angegebene Feld ignoriert.

9.5. Konfigurieren von Regeln mit pcs

Verwenden Sie den folgenden Befehl, um eine Regel zu konfigurieren. Falls Sie **score** weglassen, wird standardmäßig INFINITY verwendet. Falls Sie die **id** weglassen, wird eine ID aus der *constraint_id* generiert. Der *rule_type* sollte **expression** oder **date_expression** sein.

```
pcs constraint rule add constraint_id [rule_type] [score=score
[id=rule_id] expression|date_expression|date_spec options
```

Verwenden Sie den folgenden Befehl, um eine Regel zu entfernen. Falls die Regel, die Sie entfernen, die letzte Regel in der Beschränkung ist, wird die Beschränkung entfernt.

```
pcs constraint rule remove rule_id
```

9.6. Beispiel für zeitbasierte Ausdrücke

Der folgende Befehl konfiguriert einen Ausdruck, der **true** ergibt, falls der aktuelle Zeitpunkt im Jahr 2005 liegt.

```
# pcs constraint location Webserver rule score=INFINITY date-spec
years=2005
```

Der folgende Befehl konfiguriert einen Ausdruck, der **true** ergibt zwischen 9 und 17 Uhr, montags bis freitags. Beachten Sie, dass der angegebene Wert für die Stunden (16) bis 16:59:59 Uhr reicht, da der numerische Wert für die Stunde bis dahin übereinstimmt.

```
# pcs constraint location Webserver rule score=INFINITY date-spec
hours="9-16" weekdays="1-5"
```

Der folgende Befehl konfiguriert einen Ausdruck, der **true** ergibt, falls es an einem Freitag, den 13. einen Vollmond gibt.

```
# pcs constraint location Webserver rule date-spec weekdays=5
monthdays=13 moon=4
```

9.7. Verwenden von Regeln zum Bestimmen von Ressourcenstandorten

Mithilfe des folgenden Befehls können Sie eine Regel verwenden, um den Standort einer Ressource zu bestimmen.

```
pcs resource constraint location resource_id rule [rule_id]
[role=master|slave] [score=score expression]
```

Der Ausdruck *expression* kann einer der folgenden sein:

- **defined|not_defined *attribute***
- ***attribute* lt|gt|lte|gte|eq|ne *value***
- **date [start=*start* [end=*end* operation=gt|lt|in-range**
- **date-spec *date_spec_options***

Kapitel 10. Pacemaker-Cluster-Eigenschaften

Cluster-Eigenschaften legen fest, wie sich der Cluster in bestimmten Situationen während des Cluster-Betriebs verhält.

- Tabelle 10.1, „Cluster-Eigenschaften“ beschreibt die Optionen der Cluster-Eigenschaften.
- Abschnitt 10.2, „Festlegen und Entfernen von Cluster-Eigenschaften“ beschreibt, wie Cluster-Eigenschaften festgelegt werden.
- Abschnitt 10.3, „Abfragen der Einstellungen für Cluster-Eigenschaften“ beschreibt, wie die derzeit festgelegten Cluster-Eigenschaften angezeigt werden.

10.1. Überblick über Cluster-Eigenschaften und -Optionen

Tabelle 10.1, „Cluster-Eigenschaften“ fasst die Pacemaker-Cluster-Eigenschaften zusammen und zeigt die Standardwerte und die zulässigen Werte für diese Eigenschaften.

Anmerkung

Zusätzlich zu den in dieser Tabelle beschriebenen Eigenschaften gibt es weitere Cluster-Eigenschaften, die in der Cluster-Software bereitgestellt werden. Für diese Eigenschaften wird jedoch empfohlen, dass die Standardwerte nicht verändert werden.

Tabelle 10.1. Cluster-Eigenschaften

Option	Standard	Beschreibung
batch-limit	30	Die Anzahl der Jobs, die von der Transition Engine (TE) gleichzeitig ausgeführt werden dürfen. Der empfohlene Wert hängt von der Geschwindigkeit und der Auslastung des Netzwerks und der Cluster-Knoten ab.
migration-limit	-1 (unbegrenzt)	Die Anzahl der Migrationsjobs, die von der TE gleichzeitig auf einem Knoten ausgeführt werden dürfen.
no-quorum-policy	stop	Vorgehensweise, falls der Cluster kein Quorum hat. Zulässige Werte: * ignore – Mit der Ressourcenverwaltung fortfahren * freeze – Mit der Ressourcenverwaltung fortfahren, aber keine Ressourcen auf Knoten in der betroffenen Partition wiederherstellen * stop – Alle Ressourcen in der betroffenen Cluster-Partition stoppen * suicide – Alle Knoten in der betroffenen Cluster-Partition abgrenzen
symmetric-cluster	true	Legt fest, ob Ressourcen standardmäßig auf einem beliebigen Knoten laufen können.

Option	Standard	Beschreibung
stonith-enabled	true	Legt fest, dass ausgefallene Knoten sowie Knoten mit Ressourcen, die nicht gestoppt werden können, abgegrenzt werden sollen. Zur Sicherung Ihrer Daten muss dies auf **true** eingestellt werden. Falls **true** oder nicht festgelegt, wird der Cluster keine Ressourcen starten, bis nicht eine oder mehr STONITH-Ressourcen konfiguriert wurden.
stonith-action	reboot	An das STONITH-Gerät zu sendende Aktion. Zulässige Werte: **reboot**, **off**. Der Wert **poweroff** ist ebenfalls erlaubt, wird jedoch nur für veraltete Geräte verwendet.
cluster-delay	60s	Verzögerung beim Senden über das Netzwerk (Ausführung der Aktion ausgenommen). Der empfohlene Wert hängt von der Geschwindigkeit und der Auslastung des Netzwerks und der Cluster-Knoten ab.
stop-orphan-resources	true	Legt fest, ob gelöschte Ressourcen gestoppt werden sollen.
stop-orphan-actions	true	Legt fest, ob gelöschte Aktionen abgebrochen werden sollen.
start-failure-is-fatal	true	Legt fest, ob ein fehlgeschlagener Start als fataler Fehler für die Ressource gehandhabt wird. Falls auf **false** gesetzt, verwendet der Cluster stattdessen den **failcount** und den Wert für **migration-threshold** der Ressource. Informationen über das Einstellen der Option **migration-threshold** für eine Ressource finden Sie in Abschnitt 7.2, „Verlegen von Ressourcen wegen Ausfall“.
pe-error-series-max	-1 (alle)	Die Anzahl der PE-Eingaben, die in Fehlern beim Speichern resultieren. Verwendet zur Fehlermeldung.
pe-warn-series-max	-1 (alle)	Die Anzahl der PE-Eingaben, die in Warnungen beim Speichern resultieren. Verwendet zur Fehlermeldung.
pe-input-series-max	-1 (alle)	Die Anzahl normaler PE-Eingaben zum Speichern. Verwendet zur Fehlermeldung.
cluster-infrastructure		Der Nachrichten-Stack, auf dem Pacemaker derzeit läuft. Verwendet für Informations- und Diagnosezwecke; nicht konfigurierbar.
dc-version		Version von Pacemaker auf dem Designated Controller (DC) des Clusters. Verwendet für Diagnosezwecke; nicht konfigurierbar.
last-lrm-refresh		Letzte Aktualisierung des lokalen Ressourcenverwalters, angegeben in Sekunden seit der Epoche. Verwendet für Diagnosezwecke; nicht konfigurierbar.
cluster-recheck-interval	60	Polling-Intervall für zeitbasierte Änderungen an Optionen, Ressourcenparametern oder Beschränkungen. Zulässige Werte: Null deaktiviert das Polling, positive Ganzzahlwerte sind ein Intervall in Sekunden (es sei denn, andere SI-Einheiten werden festgelegt wie z. B. 5min).

Option	Standard	Beschreibung
default-action-timeout	20s	Wert für die Zeitüberschreitung einer Pacemaker-Aktion. Die Einstellung für eine Operation in einer Ressource selbst hat immer Vorrang vor dem Standardwert, der als Cluster-Option festgelegt wurde.
maintenance-mode	false	Dieser Wartungsmodus versetzt den Cluster in einen Wartezustand, in dem der Cluster keine Dienste startet oder stoppt, sofern nicht dazu aufgefordert. Sobald der Wartungsmodus abgeschlossen ist, führt der Cluster eine Zustandsprüfung aller Dienste durch und stoppt oder startet jegliche nötige Dienste.
shutdown-escalation	20min	Die Zeit, nach der ein Versuch zum ordnungsgemäßen Herunterfahren aufgegeben und einfach beendet werden soll. Nur zur fortgeschrittenen Verwendung.
stonith-timeout	60s	Zeitspanne, die auf den Abschluss einer STONITH-Aktion gewartet werden soll.
stop-all-resources	false	Legt fest, ob der Cluster alle Ressourcen stoppen soll.
default-resource-stickiness	5000	Legt fest, wie stark eine Ressource an ihrem derzeitigen Standort bleiben möchte. Es wird empfohlen, diesen Wert als Ressourcen-/Operationsstandard festzulegen statt als Cluster-Option.
is-managed-default	true	Legt fest, ob der Cluster eine Ressource starten und stoppen darf. Es wird empfohlen, diesen Wert als Ressourcen-/Operationsstandard festzulegen statt als Cluster-Option.
enable-acl	false	(Red Hat Enterprise Linux 6.6 und höher) Legt fest, ob der Cluster Zugriffssteuerungslisten verwenden kann, die mit dem Befehl **pcs acl** eingestellt werden.

10.2. Festlegen und Entfernen von Cluster-Eigenschaften

Verwenden Sie den folgenden **pcs**-Befehl, um den Wert für eine Cluster-Eigenschaft festzulegen.

```
pcs property set property=value
```

Um beispielsweise den Wert für **symmetric-cluster** auf **false** zu setzen, führen Sie den folgenden Befehl aus.

```
# pcs property set symmetric-cluster=false
```

Mithilfe des folgenden Befehls können Sie eine Cluster-Eigenschaft aus der Konfiguration entfernen.

```
pcs property unset property
```

Alternativ können Sie eine Cluster-Eigenschaft aus einer Konfiguration entfernen, indem Sie das Wertefeld des Befehls **pcs property set** leer lassen. Dadurch wird die Eigenschaft auf ihren Standardwert zurückgesetzt. Falls Sie beispielsweise die Eigenschaft **symmetric-cluster** bislang auf **false** eingestellt hatten, dann entfernt der folgende Befehl diesen Wert aus der Konfiguration und setzt den Wert für **symmetric-cluster** stattdessen auf seinen Standardwert **true**.

```
# pcs property set symmetic-cluster=
```

10.3. Abfragen der Einstellungen für Cluster-Eigenschaften

Wenn Sie den **pcs**-Befehl zum Anzeigen von Werten der verschiedenen Cluster-Komponenten verwenden, können in den meisten Fällen die Befehle **pcs list** oder **pcs show** austauschbar verwendet werden. In den folgenden Beispielen ist **pcs list** das verwendete Format zum Anzeigen einer umfassenden Liste aller Einstellungen für mehr als eine Eigenschaft, während **pcs show** das verwendete Format zum Anzeigen der Werte einer bestimmten Eigenschaft ist.

Verwenden Sie den folgenden **pcs**-Befehl, um die Werte der Eigenschaften anzuzeigen, die für den Cluster eingestellt wurden.

```
pcs property list
```

Verwenden Sie den folgenden Befehl, um alle Werte der Eigenschaften anzuzeigen, die für den Cluster eingestellt wurden, einschließlich jener Standardwerte, die nicht ausdrücklich festgelegt wurden.

```
pcs property list --all
```

Verwenden Sie den folgenden Befehl, um den derzeitigen Wert für eine bestimmte Cluster-Eigenschaft anzuzeigen.

```
pcs property show property
```

Um beispielsweise den derzeitigen Wert der Eigenschaft **cluster-infrastructure** anzuzeigen, führen Sie den folgenden Befehl aus:

```
# pcs property show cluster-infrastructure
Cluster Properties:
 cluster-infrastructure: cman
```

Zu Informationszwecken können Sie mithilfe des folgenden Befehls eine Liste aller Standardwerte der Eigenschaften anzeigen, ungeachtet dessen, ob diesen ein davon abweichender Wert zugewiesen wurde.

```
pcs property [list|show] --defaults
```

Cluster-Erstellung in Red Hat Enterprise Linux Release 6.5 und Red Hat Enterprise Linux Release 6.6

Die Konfiguration eines Red Hat High Availability Clusters in Red Hat Enterprise Linux 6.6 mit Pacemaker erfordert eine andere Reihe von Konfigurationstools mit einer anderen administrativen Oberfläche als die Konfiguration eines Clusters in Red Hat Enterprise Linux 6 mit **rgmanager**. Abschnitt A.1, „Cluster-Erstellung mit rgmanager und mit Pacemaker“ fasst die Konfigurationsunterschiede zwischen den verschiedenen Cluster-Komponenten zusammen.

Die Red Hat Enterprise Linux 6.6 Release bietet einige neue Features zur Cluster-Konfiguration mit Pacemaker. Abschnitt A.2, „Cluster-Erstellung mit Pacemaker in Red Hat Enterprise Linux Release 6.5 und Red Hat Enterprise Linux Release 6.6“ zeigt einige kleine Konfigurationsunterschiede zwischen der **pcs**-Unterstützung in der Red Hat Enterprise Linux Release 6.5 und der **pcs**-Unterstützung in der Red Hat Enterprise Linux Release 6.6.

A.1. Cluster-Erstellung mit rgmanager und mit Pacemaker

Tabelle A.1, „Vergleich der Cluster-Konfiguration mit rgmanager und mit Pacemaker“ liefert eine vergleichende Übersicht über die Konfiguration der Komponenten eines Clusters bei der Verwendung von **rgmanager** und bei der Verwendung von Pacemaker in der Red Hat Enterprise Linux Release 6.6.

Tabelle A.1. Vergleich der Cluster-Konfiguration mit rgmanager und mit Pacemaker

Konfigurationskomponente	rgmanager	Pacemaker
Cluster-Konfigurationsdatei	Die Cluster-Konfigurationsdatei auf jedem Knoten lautet **cluster.conf**, die direkt bearbeitet werden kann, falls gewünscht. Verwenden Sie andernfalls die **luci**- oder **ccs**-Benutzerschnittstellen zum Definieren der Cluster-Konfiguration.	Die Cluster- und Pacemaker-Konfigurationsdateien lauten **cluster.conf** und **cib.xml**. Bearbeiten Sie diese Dateien nicht direkt, sondern verwenden Sie stattdessen die **pcs**-Benutzerschnittstelle.
Netzwerkkonfiguration	Konfigurieren Sie IP-Adressen und SSH, bevor Sie den Cluster konfigurieren.	Konfigurieren Sie IP-Adressen und SSH, bevor Sie den Cluster konfigurieren.
Tools zur Cluster-Konfiguration	**luci**, **ccs**-Befehl, manuelles Bearbeiten der **cluster.conf**-Datei.	**pcs**
Installation	Installieren Sie **rgmanager** (wodurch ebenfalls dessen Abhängigkeiten installiert werden, darunter **ricci**, **luci** und die Ressourcen- und Fencing-Agents). Falls nötig, installieren Sie **lvm2-cluster** und **gfs2-utils**.	Installieren Sie **pacemaker**, **cman**, **pcs** und die von Ihnen benötigten Ressourcen- und Fencing-Agents. Falls nötig, installieren Sie **lvm2-cluster** und **gfs2-utils**.

Konfigurationskomponente	rgmanager	Pacemaker
Starten der Cluster-Dienste	Starten und aktivieren Sie die Cluster-Dienste mit dem folgenden Verfahren: 1. Starten Sie **rgmanager**, **cman** und falls nötig **clvmd** und **gfs2**. 2. Starten Sie **ricci** und zudem Sie **luci**, falls Sie die **luci**-Oberfläche verwenden. 3. Führen Sie **chkconfig on** für die benötigten Dienste aus, damit diese auf jedem relevanten Runlevel starten. Alternativ können Sie **ccs --start** ausführen, um die Cluster-Dienste zu starten und zu aktivieren.	Starten und aktivieren Sie die Cluster-Dienste mit dem folgenden Verfahren: 1. Führen Sie auf jedem Knoten **service pcsd start** aus und anschließend **service pcsd enable**, um **pcsd** zum Starten zur Laufzeit zu konfigurieren. 2. Führen Sie auf einem Knoten im Cluster **pcs cluster start --all** aus, um **cman** und **pacemaker** zu starten.
Zugriffssteuerung für Konfigurationstools	Für **luci** gilt, dass der root-Benutzer oder ein Benutzer mit **luci**-Berechtigungen auf **luci** zugreifen kann. Jeglicher Zugriff erfordert das **ricci**-Passwort für den Knoten.	Es gibt keine grafische Konfigurationsoberfläche.
Cluster-Erstellung	Verwenden Sie **luci** oder **ccs** oder bearbeiten Sie die **cluster.conf**-Datei direkt, um den Cluster zu benennen und um zu definieren, welche Knoten im Cluster enthalten sein sollen.	Verwenden Sie den Befehl **pcs cluster setup**, um den Cluster zu benennen und um Knoten zum Cluster hinzuzufügen.
Verbreiten der Cluster-Konfiguration auf alle Knoten	Bei der Konfiguration eines Clusters mit **luci** geschieht die Verbreitung automatisch. Verwenden Sie mit **ccs** die Option **--sync**. Sie können auch den Befehl **cman_tool version -r** verwenden.	Die Verbreitung der Cluster- und Pacemaker-Konfigurationsdateien **cluster.conf** und **cib.xml** geschieht automatisch bei der Einrichtung des Clusters und beim Hinzufügen einer Ressource.

Konfigurationskomponente	rgmanager	Pacemaker
Globale Cluster-Eigenschaften	**rgmanager** unterstützt die folgenden Features: * Sie können das System so konfigurieren, dass das System die zu verwendende Multicast-Adresse für das IP-Multicasting im Cluster-Netzwerk wählt. * Falls IP-Multicasting nicht verfügbar ist, können Sie den UDP-Unicast Übertragungsmechanismus verwenden. * Sie können einen Cluster zur Verwendung des RRP-Protokolls konfigurieren.	Pacemaker unterstützt die folgenden Features für einen Cluster: * Sie können für den Cluster eine **no-quorum-policy** festlegen, um anzugeben, was das System tun soll, falls der Cluster kein Quorum hat. * Weitere konfigurierbare Cluster-Eigenschaften finden Sie in Tabelle 10.1, „Cluster-Eigenschaften“.
Protokollierung	Sie können eine globale und eine Daemon-spezifische Protokollkonfiguration festlegen.	Siehe **/etc/sysconfig/pacemaker**-Datei für Informationen über die manuelle Konfiguration der Protokollierung.
Prüfen des Clusters	Die Cluster-Prüfung erfolgt mit **luci** und mit **ccs** automatisch anhand des Cluster-Schemas. Der Cluster wird automatisch beim Start geprüft.	Der Cluster wird automatisch beim Start geprüft oder Sie können den Cluster mit **pcs cluster verify** selbst prüfen.
Quorum in 2-Knoten-Clustern	Für einen 2-Knoten-Cluster können Sie konfigurieren, wie das System das Quorum bestimmt: * Konfigurieren Sie einen Quorum-Datenträger * Verwenden Sie **ccs** oder bearbeiten Sie die Datei **cluster.conf**, um **two_node=1** und **expected_votes=1** festzulegen, um es einem einzigen Knoten zu ermöglichen, das Quorum zu bewahren.	**pcs** fügt die notwendigen Optionen für einen 2-Knoten-Cluster automatisch zu **cman** hinzu.
Cluster-Status	In **luci** ist der aktuelle Status des Clusters in den verschiedenen Komponenten der Oberfläche sichtbar und kann aktualisiert werden. Mithilfe der Option **--gethost** des **ccs**-Befehls können Sie die aktuelle Konfigurationsdatei sehen. Sie können den **clustat**-Befehl verwenden, um den Cluster-Status anzuzeigen.	Sie können den derzeitigen Status des Clusters mit dem Befehl **pcs status** anzeigen.

Konfigurationskomponente	rgmanager	Pacemaker
Ressourcen	Sie können Ressourcen mit definierten Typen hinzufügen und ressourcenspezifische Eigenschaften konfigurieren, indem Sie **luci** oder den **ccs**-Befehl verwenden oder die **cluster.conf**-Konfigurationsdatei manuell bearbeiten.	Sie können Ressourcen mit definierten Typen hinzufügen und ressourcenspezifische Eigenschaften konfigurieren mit dem Befehl **pcs resource create**. Allgemeine Informationen über die Konfiguration von Cluster-Ressourcen mit Pacemaker finden Sie in Kapitel 5, *Konfigurieren von Cluster-Ressourcen*.
Ressourcenverhalten, Gruppierung und Start-/Stoppreihenfolge	Definieren Sie Cluster-*Dienste*, um zu konfigurieren, wie Ressourcen miteinander interagieren.	Pacemaker bietet mit Ressourcengruppen eine schnelle Methode, um eine Gruppe von Ressourcen zu definieren, die zusammen platziert und in bestimmter Reihenfolge gestartet und gestoppt werden müssen. Darüber hinaus definieren Sie das Verhalten und die Interaktion von Ressourcen wie folgt: * Sie können einige Aspekte des Ressourcenverhaltens als Ressourcenoptionen festlegen. * Sie können Standortbeschränkungen auferlegen, um festzulegen, auf welchen Knoten eine Ressource laufen darf. * Sie können Reihenfolgebeschränkungen verwenden, um die Reihenfolge festzulegen, in der die Ressourcen laufen. * Sie können relative Standortbeschränkungen verwenden, um festzulegen, dass der Standort einer Ressource vom Standort einer anderen Ressource abhängt. Weitere Informationen über dieses Thema finden Sie in Kapitel 5, *Konfigurieren von Cluster-Ressourcen*.

Konfigurationskomponente	rgmanager	Pacemaker
Ressourcenverwaltung: Verlegen, Starten und Stoppen von Ressourcen	Mithilfe von **luci** können Sie Cluster, einzelne Cluster-Knoten und Cluster-Dienste verwalten. Mithilfe des **ccs**-Befehls können Sie Cluster verwalten. Mithilfe des **clusvadm**-Befehls können Sie Cluster-Dienste verwalten.	Mithilfe des Befehls **pcs cluster standby** können Sie einen Knoten vorübergehend deaktivieren, sodass dieser keine Ressourcen hosten kann und die Ressourcen auf einen anderen Knoten migrieren. Mithilfe des Befehls **pcs resource disable** können Sie eine Ressource stoppen.
Vollständiges Entfernen einer Cluster-Konfiguration	Mithilfe von **luci** können Sie alle Knoten in einem Cluster zum Löschen auswählen, um diesen Cluster vollständig zu löschen. Sie können auch die **cluster.conf**-Datei von jedem Knoten im Cluster entfernen.	Mithilfe des Befehls **pcs cluster destroy** können Sie eine Cluster-Konfiguration von einem Knoten entfernen.
Aktive Ressourcen auf mehreren Knoten, aktive Ressourcen auf mehreren Knoten in mehreren Modi	Keine Entsprechung	Mit Pacemaker können Sie Ressourcen klonen, sodass diese auf mehreren Knoten laufen können, und Sie können die geklonten Ressourcen als Master- bzw. Slave-Ressource definieren, sodass diese in mehreren Modi laufen können. Weitere Informationen über geklonte Ressourcen und Master-/Slave-Ressourcen finden Sie in Kapitel 8, *Erweiterte Ressourcentypen*.
Fencing – einzelnes Fencing-Gerät pro Knoten	Erstellen Sie Fencing-Geräte global oder lokal und fügen Sie diese zu den Knoten hinzu. Sie können Werte für **post-fail delay** und **post-join delay** für den gesamten Cluster definieren.	Erstellen Sie ein Fencing-Gerät für jeden Knoten mit dem Befehl **pcs stonith create**. Geräte, die mehrere Knoten abgrenzen können, brauchen Sie nur einmal zu definieren statt separat für jeden Knoten. Sie können auch **pcmk_host_map** definieren, um mit einem einzigen Befehl Fencing-Geräte für alle Knoten zu konfigurieren. Weitere Informationen über **pcmk_host_map** finden Sie in Tabelle 4.1, „Allgemeine Eigenschaften von Fencing-Geräten“. Sie können den Wert für den **stonith-timeout** für den gesamten Cluster definieren.
Mehrere (Backup-)Fencing-Geräte pro Knoten	Definieren Sie Backup-Geräte mit **luci**, dem **ccs**-Befehl oder durch direktes Bearbeiten der **cluster.conf**-Datei.	Konfigurieren Sie Fencing-Level.

A.2. Cluster-Erstellung mit Pacemaker in Red Hat Enterprise Linux Release 6.5 und Red Hat Enterprise Linux Release 6.6

Release 6.5 und Red Hat Enterprise Linux Release 6.6

Um in Red Hat Enterprise Linux 6.5 einen Pacemaker-Cluster zu erstellen, müssen Sie den Cluster erstellen und die Cluster-Dienste auf jedem Knoten im Cluster starten. Um beispielsweise einen Cluster namens **my_cluster** zu erstellen, der aus den Knoten **z1.example.com** und **z2.example.com** besteht, und Cluster-Dienste auf diesen Knoten zu starten, führen Sie die folgenden Befehle sowohl auf **z1.example.com** als auch auf **z2.example.com** aus.

```
[root@rhel6.5]# pcs cluster setup --name my_cluster z1.example.com
z2.example.com
[root@rhel6.5]# pcs cluster start
```

In Red Hat Enterprise Linux 6.6 führen Sie den Befehl zur Cluster-Erstellung auf einem Knoten im Cluster aus. Der folgende Befehl, der nur auf einem Knoten ausgeführt wird, erstellt einen Cluster namens **my_cluster**, der aus den Knoten **z1.example.com** und **z2.example.com** besteht, und startet Cluster-Dienste auf diesen Knoten.

```
[root@rhel6.6]# pcs cluster setup --start --name my_cluster
z1.example.com z2.example.com
```

Konfigurationsbeispiel unter Verwendung der pcs-Befehle

Dieser Anhang liefert eine schrittweise Anleitung zur Konfiguration eines Red Hat Enterprise Linux High Availability Add-On Clusters mit zwei Knoten unter Verwendung des **pcs**-Befehls in der Red Hat Enterprise Linux Release 6.6 und höher. Auch die Konfiguration eines Apache-Webservers in diesem Cluster wird erläutert.

Für die Konfiguration des in diesem Kapitel beschriebenen Clusters benötigt Ihr System die folgenden Komponenten:

- 2 Knoten, aus denen der Cluster gebildet wird. In diesem Beispiel werden die Knoten **z1.example.com** und **z2.example.com** verwendet.
- Netzwerk-Switches für das private Netzwerk, die zur Kommunikation zwischen den Cluster-Knoten und anderer Cluster-Hardware wie z. B. Network Power Switches und Fibre Channel Switches notwendig sind.
- Ein Power-Fencing-Gerät für jeden Knoten im Cluster. Dieses Beispiel verwendet zwei Ports des APC Power Switches mit dem Hostnamen **zapc.example.com**.

B.1. Erste Systemeinrichtung

Dieser Abschnitt beschreibt die ersten Einrichtungsschritte für das System, das Sie zur Erstellung des Clusters nutzen werden.

B.1.1. Installieren der Cluster-Software

Führen Sie die folgenden Schritte aus, um die Cluster-Software zu installieren.

1. Vergewissern Sie sich, dass **pacemaker**, **cman** und **pcs** installiert sind.

```
yum install -y pacemaker cman pcs
```

2. Führen Sie nach der Installation den folgenden Befehl auf allen Knoten im Cluster aus, um zu verhindern, dass **corosync** ohne **cman** startet.

```
# chkconfig corosync off
```

3. Falls Sie sichergehen möchten, dass **cman** auch ohne Quorum bei mehr als zwei Knoten im Cluster vollständig startet, führen Sie den folgenden Befehl aus.

```
# sed -i.sed
"s/.*CMAN_QUORUM_TIMEOUT=.*/CMAN_QUORUM_TIMEOUT=0/g"
/etc/sysconfig/cman
```

B.1.2. Erstellen und Starten des Clusters

Dieser Abschnitt zeigt die notwendigen Schritte zum Erstellen des anfänglichen Clusters, auf dem Sie dann die Cluster-Ressourcen konfigurieren.

1. Um **pcs** zur Konfiguration der Knoten und zur Kommunikation der Knoten untereinander zu verwenden, müssen Sie auf jedem Knoten ein Passwort für die Benutzer-ID **hacluster** (den **pcs**-Administrationsaccount) festlegen. Es empfiehlt sich, auf allen Knoten dasselbe Passwort für den Benutzer **hacluster** zu wählen.

```
# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

2. Bevor der Cluster konfiguriert werden kann, muss der **pcsd**-Daemon gestartet werden. Dieser Daemon arbeitet zusammen mit dem **pcs**-Befehl, um die Konfiguration auf den Knoten im Cluster zu verwalten.

   Führen Sie auf jedem Knoten im Cluster die folgenden Befehle aus, um den **pcsd**-Dienst zu starten und **pcsd** beim Systemstart zu aktivieren.

```
# service pcsd start
# service pcsd enable
```

3. Authentifizieren Sie den **pcs**-Benutzer **hacluster** für jeden Knoten im Cluster auf dem Knoten, auf dem Sie **pcs** ausführen werden.

   Der folgende Befehl authentifiziert den Benutzer **hacluster** auf **z1.example.com** für beide Knoten in unserem Zwei-Knoten-Cluster, **z1.example.com** und **z2.example.com**.

```
root@z1 ~]# pcs cluster auth z1.example.com z2.example.com
Username: hacluster
Password:
z1.example.com: Authorized
z2.example.com: Authorized
```

4. Führen Sie den folgenden Befehl auf **z1.example.com** aus, um den Zwei-Knoten-Cluster **mycluster** zu erstellen, der aus den Knoten **z1.example.com** und **z2.example.com** besteht. Dies überträgt die Cluster-Konfigurationsdateien auf beide Knoten im Cluster. Dieser Befehl enthält die Option **--start**, wodurch die Cluster-Dienste auf beiden Knoten im Cluster gestartet werden.

```
[root@z1 ~]# pcs cluster setup --start --name my_cluster \
z1.example.com z2.example.com
z1.example.com: Succeeded
z1.example.com: Starting Cluster...
z2.example.com: Succeeded
z2.example.com: Starting Cluster...
```

5. Optional können Sie die Cluster-Dienste so konfigurieren, dass diese auf beiden Knoten im Cluster starten, sobald der Knoten gebootet wird.



Anmerkung

Für Ihre Umgebung können Sie diesen Schritt jedoch auch überspringen und somit die Cluster-Dienste deaktiviert lassen. Dies ermöglicht Ihnen im Falle eines Knotenausfalls eine Suche und Bereinigung von Fehlern, bevor der Knoten wieder dem Cluster beitritt. Falls Sie die Cluster-Dienste deaktiviert lassen und Sie einen Knoten neu starten, müssen Sie die Cluster-Dienste manuell durch Ausführen des Befehls **pcs cluster start** auf diesem Knoten starten.

```
# pcs cluster enable --all
```

Sie können den derzeitigen Status des Clusters mit dem Befehl **pcs cluster status** anzeigen.

```
[root@z1 ~]# pcs cluster status
Cluster Status:
 Last updated: Thu Jul 25 13:01:26 2013
 Last change: Thu Jul 25 13:04:45 2013 via crmd on z2.example.com
 Stack: corosync
 Current DC: z2.example.com (2) - partition with quorum
 Version: 1.1.10-5.el7-9abe687
 2 Nodes configured
 0 Resources configured
```

B.2. Fencing-Konfiguration

Sie müssen ein Fencing-Gerät für jeden Knoten im Cluster konfigurieren. Allgemeine Informationen über die Konfiguration von Fencing-Geräten finden Sie in Kapitel 4, *Fencing: Konfigurieren von STONITH*.

Anmerkung

Bei der Konfiguration eines Fencing-Geräts sollten Sie darauf achten, dass Ihr Fencing-Gerät nicht an dieselbe Stromversorgung angeschlossen ist wie der Knoten, für den es zuständig ist.

Dieses Beispiel verwendet den APC Power Switch mit dem Hostnamen **zapc.example.com** zur Abgrenzung der Knoten und es verwendet den Fencing-Agent **fence_apc_snmp**. Da beide Knoten von demselben Fencing-Agent abgegrenzt werden, können Sie beide Fencing-Geräte als eine einzige Ressource konfigurieren mithilfe der Optionen **pcmk_host_map** und **pcmk_host_list**.

Sie erstellen ein Fencing-Gerät, indem Sie das Gerät mithilfe des Befehls **pcs stonith create** als **stonith**-Ressource konfigurieren. Der folgende Befehl konfiguriert eine **stonith**-Ressource namens **myapc**, die den Fencing-Agent **fence_apc_snmp** für die Knoten **z1.example.com** und **z2.example.com** verwendet. Die Option **pcmk_host_map** weist **z1.example.com** zu Port 1 und **z2.example.com** zu Port 2 zu. Der Benutzername und das Passwort für das APC-Gerät lauten beide **apc**. Standardmäßig nutzt dieses Gerät ein Prüfintervall von 60 Sekunden auf jedem Knoten.

Beachten Sie, dass Sie zur Angabe des Hostnamens für die Knoten auch eine IP-Adresse verwenden können.

```
[root@z1 ~]# pcs stonith create myapc fence_apc_snmp params \
```

```
ipaddr="zapc.example.com"
pcmk_host_map="z1.example.com:1;z2.example.com:2" \
pcmk_host_check="static-list"
pcmk_host_list="z1.example.com,z2.example.com" \
login="apc" passwd="apc"
```



Anmerkung

Wenn Sie ein **stonith**-Gerät mit **fence_apc_snmp** konfigurieren, sehen Sie gegebenenfalls die folgende Warnmeldung, die Sie jedoch problemlos ignorieren können:

```
Warning: missing required option(s): 'port, action' for resource
type: stonith:fence_apc_snmp
```

Der folgende Befehl zeigt die Parameter eines vorhandenen STONITH-Geräts.

```
[root@rh7-1 ~]# pcs stonith show myapc
 Resource: myapc (class=stonith type=fence_apc_snmp)
  Attributes: ipaddr=zapc.example.com
pcmk_host_map=z1.example.com:1;z2.example.com:2 pcmk_host_check=static-
list pcmk_host_list=z1.example.com,z2.example.com login=apc passwd=apc
  Operations: monitor interval=60s (myapc-monitor-interval-60s)
```

B.3. Konfigurieren eines Apache-Webservers in einem Red Hat High Availability Cluster mithilfe des pcs-Befehls

Dieser Abschnitt beschreibt die Konfiguration eines Apache-Webservers in einem Red Hat Enterprise Linux High Availability Add-On Cluster mit zwei Knoten unter Verwendung von **pcs** zur Konfiguration von Cluster-Ressourcen. In diesem Beispiel greifen Clients über eine Floating-IP-Adresse auf den Apache-Webserver zu. Der Webserver läuft auf einem der zwei Knoten im Cluster. Falls der Knoten, auf dem der Webserver läuft, ausfällt, startet der Webserver auf dem zweiten Knoten im Cluster ohne große Serviceunterbrechung.

Für dieses Beispiel benötigt Ihr System die folgenden Komponenten:

- Einen Red Hat High Availability Cluster bestehend aus zwei Knoten mit Power-Fencing für jeden Knoten. Dieses Verfahren verwendet das Cluster-Beispiel aus Abschnitt B.1.2, „Erstellen und Starten des Clusters“.
- Eine öffentliche, virtuelle IP-Adresse, erforderlich für den Apache-Webserver.
- Gemeinsam verwendeten Speicherplatz für die Knoten im Cluster mittels iSCSI oder Fibre Channel.

Der Cluster wird mit einer Apache-Ressourcengruppe konfiguriert, welche die Cluster-Komponenten enthält, die vom Webserver benötigt werden: eine LVM-Ressource, eine Dateisystemressource, eine IP-Adressressource und eine Webserverressource. Diese Ressourcengruppe kann im Falle eines Ausfalls von einem Knoten im Cluster auf einen anderen verlegt werden, sodass beide Knoten den Webserver ausführen können. Bevor Sie die Ressourcengruppe für diesen Cluster erstellen, führen Sie die folgenden Aufgaben durch:

1. Konfigurieren Sie ein **ext4**-Dateisystem, das auf dem logischen Datenträger **my_lv**

eingehängt ist, wie in Abschnitt B.3.1, „Konfigurieren eines LVM-Datenträgers mit einem ext4-Dateisystem“ beschrieben.

2. Konfigurieren Sie einen Webserver, wie in Abschnitt B.3.2, „Webserver-Konfiguration“ beschrieben.

3. Vergewissern Sie sich, dass nur der Cluster dazu in der Lage ist, die Datenträgergruppe zu aktivieren, die **my_lv** enthält, und dass die Datenträgergruppe beim Systemstart nicht außerhalb des Clusters aktiviert wird, wie in Abschnitt B.3.3, „Exklusive Archivierung einer Datenträgergruppe in einem Cluster“ beschrieben.

Nachdem Sie diese Aufgaben durchgeführt haben, können Sie die Ressourcengruppe und die darin enthaltenen Ressourcen erstellen, wie in Abschnitt B.3.4, „Erstellen der Ressourcen und Ressourcengruppen mit dem pcs-Befehl“ beschrieben.

B.3.1. Konfigurieren eines LVM-Datenträgers mit einem ext4-Dateisystem

Dieses Beispiel erfordert, dass Sie einen logischen LVM-Datenträger auf Speicher anlegen, der von den Knoten im Cluster gemeinsam genutzt werden kann.

Das folgende Verfahren erstellt einen logischen LVM-Datenträger und legt anschließend ein **ext4**-Dateisystem auf diesem Datenträger an. In diesem Beispiel wird die gemeinsam verwendete Partition **/dev/sdb1** zur Speicherung des physischen LVM-Datenträgers verwendet, auf dem der logische LVM-Datenträger erstellt wird.



Anmerkung

LVM-Datenträger und zugehörige Partitionen und Geräte, die von Cluster-Knoten verwendet werden, dürfen ausschließlich mit den Cluster-Knoten verbunden werden.

Da es sich bei der Partition **/dev/sdb1** um gemeinsam verwendeten Speicher handelt, führen Sie dieses Verfahren nur auf einem Knoten aus.

1. Erstellen Sie einen physischen LVM-Datenträger auf der Partition **/dev/sdb1**.

```
# pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
```

2. Erstellen Sie die Datenträgergruppe **my_vg**, die den physischen Datenträger **/dev/sdb1** umfasst.

```
# vgcreate my_vg /dev/sdb1
  Volume group "my_vg" successfully created
```

3. Erstellen Sie einen logischen Datenträger auf der Datenträgergruppe **my_vg**.

```
# lvcreate -L450 -n my_lv my_vg
  Rounding up size to full physical extent 452.00 MiB
  Logical volume "my_lv" created
```

Sie können den **lvs**-Befehl verwenden, um den logischen Datenträger anzuzeigen.

```
# lvs
```

```
  LV       VG       Attr       LSize   Pool Origin Data%  Move Log
Copy%  Convert
  my_lv    my_vg    -wi-a---- 452.00m
  ...
```

4. Legen Sie ein **ext4**-Dateisystem auf dem logischen Datenträger **my_lv** an.

```
# mkfs.ext4 /dev/my_vg/my_lv
mke2fs 1.42.7 (21-Jan-2013)
Filesystem label=
OS type: Linux
...
```

B.3.2. Webserver-Konfiguration

Die folgenden Schritte beschreiben, wie ein Apache-Webserver konfiguriert wird.

1. Vergewissern Sie sich, dass der HTTPD-Server auf jedem Knoten im Cluster installiert ist. Darüber hinaus muss das Tool **wget** auf dem Cluster installiert sein, um den Status des Apache-Webservers zu prüfen.

   Führen Sie auf jedem Knoten den folgenden Befehl aus.

```
# yum install -y httpd wget
```

2. Damit der Apache-Ressourcenagent den Status des Apache-Webservers abrufen kann, vergewissern Sie sich, dass der folgende Text auf jedem Knoten im Cluster in der Datei **/etc/httpd/conf/httpd.conf** enthalten ist und dass dieser Text nicht auskommentiert wurde. Falls dieser Text noch nicht enthalten ist, fügen Sie ihn am Ende der Datei hinzu.

```

<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

3. Erstellen Sie eine Webseite, die Apache anzeigen soll. Hängen Sie auf einem Knoten im Cluster das Dateisystem ein, das Sie in Abschnitt B.3.1, „Konfigurieren eines LVM-Datenträgers mit einem ext4-Dateisystem“ erstellt haben, erstellen Sie die Datei **index.html** auf diesem Dateisystem und hängen Sie das Dateisystem anschließend aus.

```
# mount /dev/my_vg/my_lv /var/www/
# mkdir /var/www/html
# mkdir /var/www/cgi-bin
# mkdir /var/www/error
# restorecon -R /var/www
# cat <<-END >/var/www/html/index.html
<html>
<body>Hello</body>
</html>
END
# umount /var/www
```

B.3.3. Exklusive Archivierung einer Datenträgergruppe in einem Cluster

Das folgende Verfahren konfiguriert die Datenträgergruppe derart, um sicherzustellen, dass nur der Cluster dazu in der Lage ist, die Datenträgergruppe zu aktivieren und dass die Datenträgergruppe beim Systemstart nicht außerhalb des Clusters aktiviert werden kann. Falls die Datenträgergruppe von einem System außerhalb des Clusters aktiviert wird, besteht das Risiko, die Metadaten der Datenträgergruppe zu beschädigen.

Dieses Verfahren ändert den Eintrag für **volume_list** in der Konfigurationsdatei **/etc/lvm/lvm.conf**. Die im Eintrag **volume_list** aufgeführten Datenträgergruppen dürfen automatisch auf dem lokalen Knoten außerhalb der Kontrolle des Cluster-Managers aktiviert werden. Datenträgergruppen, die mit den lokalen root- und Benutzerverzeichnissen des Knotens in Verbindung stehen, sollten in dieser Liste aufgeführt sein. Alle Datenträgergruppen, die vom Cluster-Manager verwaltet werden, müssen vom Eintrag **volume_list** ausgenommen sein. Beachten Sie, dass dieses Verfahren nicht die Verwendung von **clvmd** erfordert.

Führen Sie das folgende Verfahren auf jedem Knoten im Cluster aus.

1. Finden Sie mithilfe des folgenden Befehls heraus, welche Datenträgergruppen derzeit auf Ihrem lokalen Speicher konfiguriert sind. Sie erhalten eine Ausgabe mit den derzeit konfigurierten Datenträgergruppen. Falls Sie auf diesem Knoten in separaten Datenträgergruppen Speicherplatz zugewiesen haben für Ihre root- und Benutzerverzeichnisse, dann sehen Sie diese Datenträger in der Ausgabe, wie in diesem Beispiel.

```
# vgs --noheadings -o vg_name
  my_vg
  rhel_home
  rhel_root
```

2. Fügen Sie die Datenträgergruppen mit Ausnahme von **my_vg** (die Datenträgergruppe, die Sie eben für den Cluster definiert haben) als Einträge zu **volume_list** in der Konfigurationsdatei **/etc/lvm/lvm.conf** hinzu. Falls Sie beispielsweise Speicherplatz in einer separaten Datenträgergruppe für Ihre root- und Benutzerverzeichnisse zugewiesen haben, würden Sie die Zeile **volume_list** in der Datei **lvm.conf** auskommentieren und diese Datenträgergruppen wie folgt als Einträge zu **volume_list** hinzufügen:

```
volume_list = [ "rhel_root", "rhel_home" ]
```



Anmerkung

Falls keine lokalen Datenträgergruppen auf einem Knoten existieren, die außerhalb des Cluster-Manager aktiviert werden sollen, müssen Sie dennoch den Eintrag **volume_list** als **volume_list = []** initialisieren.

3. Erstellen Sie das Boot-Image **initramfs** neu, um zu gewährleisten, dass das Boot-Image nicht versuchen wird, eine vom Cluster gesteuerte Datenträgergruppe zu aktivieren. Aktualisieren Sie das **initramfs**-Gerät mit dem folgenden Befehl. Es kann unter Umständen bis zu einer Minute dauern, bis der Befehl abgeschlossen ist.

```
# dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

4. Starten Sie den Knoten neu.



Anmerkung

Falls Sie nach dem Booten des Knotens, auf dem Sie das Boot-Image erstellt haben, einen neuen Linux-Kernel installiert haben, dann gilt das **initrd**-Image für den Kernel, der zum Zeitpunkt der Boot-Image-Erstellung installiert war und nicht für den neuen Kernel, der nach dem Neustart des Knotens ausgeführt wird. Sie können sich vergewissern, dass das richtige **initrd**-Gerät verwendet wird, indem Sie den Befehl **uname -r** vor und nach dem Neustart ausführen, um die derzeit laufende Kernel-Release anzuzeigen. Falls die beiden Angaben nicht übereinstimmen, aktualisieren Sie die **initrd**-Datei nach dem Neustart mit dem neuen Kernel und starten Sie den Knoten neu.

5. Sobald der Knoten neu gestartet wurde, vergewissern Sie sich, ob die Cluster-Dienste auf dem Knoten wieder gestartet wurden, indem Sie den Befehl **pcs cluster status** auf dem Knoten ausführen. Falls die Meldung **Error: cluster is not currently running on this node** angezeigt wird, führen Sie folgenden Befehl aus.

```
# pcs cluster start
```

Alternativ können Sie warten, bis Sie jeden Knoten im Cluster neu gestartet haben, und dann die Cluster-Dienste auf jedem Knoten mit dem folgenden Befehl starten.

```
# pcs cluster start --all
```

B.3.4. Erstellen der Ressourcen und Ressourcengruppen mit dem pcs-Befehl

Dieser Anwendungsfall erfordert, dass Sie vier Cluster-Ressourcen erstellen. Um sicherzustellen, dass diese Ressourcen alle auf demselben Knoten laufen, werden Sie als Mitglieder der Ressourcengruppe **apachegroup** konfiguriert. Nachfolgend sehen Sie die zu erstellenden Ressourcen, angegeben in der Reihenfolge, in der sie starten werden.

1. Eine **LVM**-Ressource namens **my_lvm**, welche die LVM-Datenträgergruppe verwendet, die Sie in Abschnitt B.3.1, „Konfigurieren eines LVM-Datenträgers mit einem ext4-Dateisystem“ erstellt haben.

2. Eine **Filesystem**-Ressource namens **my_fs**, die das Dateisystemgerät **/dev/my_vg/my_lv** verwendet, das Sie in Abschnitt B.3.1, „Konfigurieren eines LVM-Datenträgers mit einem ext4-Dateisystem“ erstellt haben.

3. Eine **IPaddr2**-Ressource, die eine Floating-IP-Adresse für die **apachegroup**-Ressourcengruppe ist. Die IP-Adresse darf nicht bereits einem physischen Knoten zugewiesen sein. Falls das NIC-Gerät der **IPaddr2**-Ressource nicht angegeben wird, muss sich die Floating-IP-Adresse auf demselben Netzwerk befinden wie die statisch zugewiesenen IP-Adressen, die von den Cluster-Knoten verwendet werden. Andernfalls kann das der Floating-IP-Adresse zuzuweisende NIC-Gerät nicht richtig erkannt werden.

4. Eine **apache**-Ressource namens **Website**, welche die **index.html**-Datei und die Apache-Konfiguration verwendet, die Sie in Abschnitt B.3.2, „Webserver-Konfiguration“ definiert haben.

Das folgende Verfahren erstellt die Ressourcengruppe **apachegroup** sowie die Ressourcen, die diese Gruppe enthält. Die Ressourcen starten in der Reihenfolge, in der Sie diese zur Gruppe hinzufügen, und sie stoppen in der umgekehrten Reihenfolge, in der Sie zur Gruppe hinzugefügt wurden. Führen Sie dieses Verfahren nur auf einem Knoten im Cluster aus.

1. Der folgende Befehl erstellt die LVM-Ressource **my_lvm**. Dieser Befehl legt den Parameter **exclusive=true** fest, um sicherzustellen, dass nur der Cluster dazu in der Lage ist, den logischen LVM-Datenträger zu aktivieren. Da die Ressourcengruppe **apachegroup** noch nicht existiert, erstellt dieser Befehl diese Ressourcengruppe.

```
[root@z1 ~]# pcs resource create my_lvm LVM volgrpname=my_vg \
exclusive=true --group apachegroup
```

   Wenn Sie eine Ressource erstellen, wird die Ressource automatisch gestartet. Mithilfe des folgenden Befehls können Sie bestätigen, dass die Ressource erstellt und gestartet wurde.

```
# pcs resource show
 Resource Group: apachegroup
     my_lvm (ocf::heartbeat:LVM): Started
```

   Sie können eine einzelne Ressource manuell stoppen und starten mit den Befehlen **pcs resource disable** bzw. **pcs resource enable**.

2. Die folgenden Befehle erstellen die übrigen Ressourcen für die Konfiguration und fügen diese zur Ressourcengruppe **apachegroup** hinzu.

```
[root@z1 ~]# pcs resource create my_fs Filesystem \
device="/dev/my_vg/my_lv" directory="/var/www" fstype="ext4" --
group \
apachegroup

[root@z1 ~]# pcs resource create VirtualIP IPaddr2 ip=198.51.100.3
\
cidr_netmask=24 --group apachegroup

[root@z1 ~]# pcs resource create Website apache \
configfile="/etc/httpd/conf/httpd.conf" \
statusurl="http://127.0.0.1/server-status" --group apachegroup
```

3. Nach der Erstellung der Ressourcengruppen und der darin enthaltenen Ressourcen können Sie den Status des Clusters prüfen. Beachten Sie, dass alle vier Ressourcen auf demselben Knoten laufen.

```
[root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Wed Jul 31 16:38:51 2013
Last change: Wed Jul 31 16:42:14 2013 via crm_attribute on
z1.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
6 Resources configured


Online: [ z1.example.com z2.example.com ]

Full list of resources:
 myapc (stonith:fence_apc_snmp): Started z1.example.com
 Resource Group: apachegroup
```

```
     my_lvm (ocf::heartbeat:LVM): Started z1.example.com
     my_fs (ocf::heartbeat:Filesystem): Started z1.example.com
     VirtualIP (ocf::heartbeat:IPaddr2): Started z1.example.com
     Website (ocf::heartbeat:apache): Started z1.example.com
```

Beachten Sie, dass die Ressourcen standardmäßig nicht starten, wenn Sie kein Fencing-Gerät für Ihren Cluster konfiguriert haben wie in Abschnitt B.2, „Fencing-Konfiguration“ beschrieben.

4. Sobald der Cluster läuft, können Sie in einem Browser die IP-Adresse aufrufen, die Sie als **IPaddr2**-Ressource definiert haben, um die einfache Beispielanzeige „Hello“ zu sehen.

```
Hello
```

Falls Sie feststellen, dass die von Ihnen konfigurierten Ressourcen nicht laufen, können Sie den Befehl **pcs resource debug-start *resource*** ausführen, um die Ressourcenkonfiguration zu testen. Informationen über den Befehl **pcs resource debug-start** finden Sie im Handbuch *High Availability Add-On Reference*.

B.3.5. Testen der Ressourcenkonfiguration

In der Cluster-Statusanzeige aus Abschnitt B.3.4, „Erstellen der Ressourcen und Ressourcengruppen mit dem pcs-Befehl“ laufen alle Ressourcen auf Knoten **z1.example.com**. Sie können testen, ob die Ressourcengruppe im Fehlerfall auf Knoten **z2.example.com** ausweicht, indem Sie mittels des folgenden Verfahrens den ersten Knoten in den **Standby**-Modus versetzen, woraufhin der Knoten nicht länger dazu in der Lage sein wird, Ressourcen zu hosten.

1. Der folgende Befehl versetzt Knoten **z1.example.com** in den **Standby**-Modus.

```
root@z1 ~]# pcs cluster standby z1.example.com
```

2. Nachdem Sie den Knoten **z1** in den Standby-Modus versetzt haben, prüfen Sie den Cluster-Status. Beachten Sie, dass die Ressourcen nun alle auf **z2** laufen sollten.

```
[root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Wed Jul 31 17:16:17 2013
Last change: Wed Jul 31 17:18:34 2013 via crm_attribute on
z1.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
6 Resources configured


Node z1.example.com (1): standby
Online: [ z2.example.com ]

Full list of resources:

 myapc (stonith:fence_apc_snmp): Started z1.example.com
 Resource Group: apachegroup
```

```
    my_lvm (ocf::heartbeat:LVM): Started z2.example.com
    my_fs (ocf::heartbeat:Filesystem): Started z2.example.com
    VirtualIP (ocf::heartbeat:IPaddr2): Started z2.example.com
    Website (ocf::heartbeat:apache): Started z2.example.com
```

Die Website unter der definierten IP-Adresse sollte nach wie vor angezeigt werden, ohne Serviceunterbrechung.

3. Um den **Standby**-Modus für **z1** aufzuheben, führen Sie den folgenden Befehl aus.

```
root@z1 ~]# pcs cluster unstandby z1.example.com
```

> **Anmerkung**
>
> Das Aufheben des **Standby**-Modus allein führt nicht dazu, dass die Ressource wieder zurück auf diesen Knoten wechselt. Informationen darüber, wie Sie steuern können, auf welchen Knoten die Ressourcen laufen können, finden Sie im Kapitel über die Konfiguration von Cluster-Ressourcen im Handbuch *Red Hat High Availability Add-On Reference*.

Versionsgeschichte

Version 2.0-7.2	Fri Apr 24 2015	Hedda Peters

Deutsche Übersetzung fertiggestellt

Version 2.0-7.1	Fri Apr 24 2015	Hedda Peters

Übersetzungsdateien synchronisiert mit XML-Quellen 2.0-7

Version 2.0-7	Tue Dec 16 2014	Steven Levine

Implementierung von sort_order auf der RHEL 6 Splash-Seite.

Version 2.0-5	Thu Oct 9 2014	Steven Levine

Version für 6.6 GA-Release

Version 2.0-4	Wed Oct 8 2014	Steven Levine

Behebt: #1131544
Fügt Dokumentation für ACLs hinzu

Version 2.0-2	Wed Aug 7 2014	Steven Levine

Version für 6.6 Beta-Release

Version 2.0-1	Wed Jul 23 2014	Steven Levine

Entwurf für 6.6 Beta

Behebt: #1126896, #1126018, #986462, #1045406, #1122145, #1079340
Kleine redaktionelle und technische Änderungen.

Behebt: #1081225, #1081248, #1092720
Aktualisiert das Dokument mit Informationen über die Unterstützung der Konfigurationssynchronisation und anderer Funktionen zur Kommunikation unter den Knoten.

Version 1.1-2	Wed Nov 20 2013	Steven Levine

Version für 6.5 GA-Release

Version 0.1-4	Wed Oct 2 2013	Steven Levine

Erste Erstellung des Entwurfs für 6.5 Beta